LOCATION-ALLOCATION-ROUTING APPROACH TO SOLID WASTE COLLECTION AND DISPOSAL

By

ADELEKE, OLAWALE JOSHUA
Matric Number: CUGP120438

MARCH, 2017

LOCATION-ALLOCATION-ROUTING APPROACH TO SOLID WASTE COLLECTION AND DISPOSAL

By

ADELEKE, OLAWALE JOSHUA
B.Tech Mathematics (Ogbomoso)
M.Sc Mathematics (Ilorin)
Matric Number: CUGP120438

A THESIS SUBMITTED TO THE DEPARTMENT OF MATHEMATICS, COLLEGE OF SCIENCE AND TECHNOLOGY, COVENANT UNIVERSITY, OTA, IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF Ph.D DEGREE IN INDUSTRIAL MATHEMATICS

MARCH, 2017

ACCEPTANCE

This is to attest that this thesis is accepted in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy in Industrial Mathematics, College of Science and Technology, Covenant University, Ota.

Philip John Ainwokhai .....

Secretary, School of Postgraduate Studies Signature & Date

Prof. Samuel T. Wara .....

Dean, School of Postgraduate Studies Signature& Date

DECLARATION

I, **ADELEKE, Olawale Joshua**, (CUGP120438), declare that this research was carried out by me under the supervision of Prof. Aderemi Oluyinka Adewumi of the Department of Computer Science, School of Mathematics, Statistics and Computer Science, University of KwaZulu Natal, Westville Campus, Durban, South Africa and Prof. Samuel Azubuike Iyase of the Department of Mathematics, Covenant University, Ota, Nigeria. I attest that the thesis has not been presented either wholly or partly for the award of any degree elsewhere. All sources of data and Scholarly information used in this thesis are duly acknowledged.

ADELEKE Olawale Joshua ..... 09/03/2017 .....

Signature & Date

CERTIFICATION

We certify that this thesis titled "Location-Allocation-Routing Approach to Solid Waste Collection and Disposal" is an original work carried out by ADELEKE, Olawale Joshua, (CUGP120438), in the Department of Mathematics, College of Science and Technology, Covenant University, Ota, Ogun State, Nigeria, under the supervision of Prof. A. O. Adewumi and Prof. S. A. Iyase. We have examined and found the work acceptable for the degree of Doctor of Philosophy in Industrial Mathematics.

Prof. Aderemi O. Adewumi .....
Supervisor Signature & Date

Prof. Samuel A. Iyase .....
Co-Supervisor Signature & Date

Dr. Timothy A. Anake .....
Acting Head of Department Signature & Date

Prof. Olabode M. Bamigbola .....
External Examiner Signature & Date

Prof. Samuel T. Wara .....
Dean, School of Postgraduate Studies Signature & Date

DEDICATION

To God Almighty, the Giver and Sustainer of my life, for His abiding grace and presence all through the course of this study.

To my wife, Victoria Ajibola Adeleke, for her unwavering support towards the completion of this thesis.

To our daughter, ModupeOluwa Peace Adeleke, whose arrival during the course of my study brought additional joy.

ACKNOWLEDGMENT

I would like to acknowledge the following people for their contributions towards the success of this work.

Foremost, my profound gratitude goes the *Visioner* and Chancellor of Covenant University, Dr. David O. Oyedepo, for establishing the platform on which this research was conducted. His wealth of wisdom is indeed a source of inspiration to me and many others. May the Lord bless him with length of years and more fruitful labour in His Vineyard. The management of Covenant University and the leadership of the School of Postgraduate Studies are deeply appreciated. I thank my examiners and assessors for their great contributions and and constructive criticisms. God bless all of them.

My heartfelt appreciation goes to my supervisor, Prof. A. O. Adewumi, who used his wealth of experience to provide necessary guidance and valuable suggestions. Thank you Sir for affording me the rare opportunity to visit the School of Mathematics, Statistics and Computer Science (SMSCS) of the University of Kwazulu-Natal (UKZN), Westville Campus, Durban, South Africa. This is an experience I will cherish and appreciate forever. Sir, God bless you and reward you abundantly. Also to my Co-supervisor, Prof. S. A. Iyase, for taking time to read through the work and for providing valuable suggestions on how to improve the contents of the work.

The immediate past Acting Head of Department of Mathematics, Dr. E. A. Owoloko, current Acting Head of Department of Mathematics, Dr. T. A. Anake and other faculty and Staff of the Department of Mathematics are deeply appreciated for their constructive criticisms and suggestions in the various seminars.

Furthermore, I deeply appreciate Dr. Michael Olusanya, Dr. Martins Arasomwan, Akinyelu Ayobami, Alochukwu Alex Somto and other members of Optimization Group of the Department of Computer Science, UKZN, Westville campus, for providing necessary assistance and materials for the numerical computations in this thesis and for their various constructive and valuable contributions. Also in this class are all the members of Deeper Life Campus Fellowship, UKZN, Westville Campus, for their love and care all through my stay in UKZN. I cannot forget the warm affections of Dr. Moses and Raphel Angulu, my residence and office mates respectively, both of who are citizens of Kenya. I thank them for being my friends.

My appreciation also goes to Robert Fourer of AMPL Optimization Inc., for providing necessary assistance and materials needed for coding and running the models on AMPL. Thanks especially for answering many of my questions directly. I owe much to him for the results in this thesis.

I am totally short of words in expressing my gratitude to my parents, Dcn. & Mrs. G. O. Adeleke, and to my siblings (Jacob, Matthew, Joseph, Samson and Thomas). I cannot forget the words of encouragement and prayers of my in-laws, the Ogundele's family. I deeply appreciate them all for their unalloyed supports and understanding.

Finally, I must say a big thank you to my wife, Victoria, and daughter, Peace, for being such an ever-refreshing spring of happiness to me. I thank God for blessing me with them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACO	Ant Colony Optimization
AI	Artifficial Intelligence
AIMMS	Advanced Interactive Multidimensional Modeling System
AMPL	A Modeling Language for Mathematical Programming
ALNS	Adaptive Large Neighbourhood Search
ASCVRP	Asymmetric Capacitated Vehicle Routing Problem
BB	Bound-and-bound
C_p**MFLP**	Capacitated p-median Facility Location Problem
CCP	Capacitated Clustering Problem
CCVRP	Cummulative Capacitated Vehicle Routing Problem
CFLP	Capacitated Facility Location Problem
CVRP	Capacitated Vehicle Routing Problem
DVRP	Dynamic Vehicle Routing Problem
DVRPTW	Dynamic Vehicle Routing Problem with Time Windows
ES	Evolutionary Search
EA	Evolutionary Algorithm
FCCFLP	Fixed Cost Capacitated Facility Location Problem
FCR	Fuel Consumption Rate
FLP	Facility Location Problem
GA	Genetic Algorithm
GCM	Greedy Clustering Method
GDMC	Geographic Distance Matrix Calculator
GRASP	Greedy Randomized Adaptive Search Procedure
ILP	Integer Linear Programming
IP	Integer Programming
KBEA	knowledge-Based Evolutionary Algorithm
LAWMA	Lagos Waste Management Agency

LD	Lagrangian Dual
LGA	Local Government Area
LOF	Localized Optimization Framework
LNS	Large Neighbourhood Search
LP	Linear Programming
LR	Lagrangian Relaxation
MCLP	Maximum Covering Location Problem
MDPVRP	Multi-Depot Periodic Vehicle Routing Problem
MFLP	Multi-Facility Location Problem
MILP	Mixed Integer Linear Programming
MIP	Mixed Integer Programming
MPDVRP	Multi-Period Dynamic Vehicle Routing Problem
MRF	Material Recovery Facility
NAM	Nearest Addition Method
NP	Non-deterministic Polynomial
PRP	Path Relinking Procedure
PSO	Particle Swarm Optimization
PTSP	Periodic Traveling Salesman Problem
PVRP	Periodic Vehicle Routing Problem
PVRPIF	Periodic Vehicle Routing Problem Intermediate Facilities
PVRP-SC	Periodic Vehicle Routing Problem with Service Choice
PVRPTW	Periodic Vehicle Routing Problem with Time Windows
RSM	Response Surface Methodology
SA	Simulated Annealing
SCLP	Set Covering Location Problems
SCVRP	Symmetric Capacitated Vehicle Routing Problem
SFLP	Single Facility Location Problems
SO	Sub-gradient Optimization

SR Solution Representation

SR-GCWS Simulated Routing via the Generalized Clarke and Wright Savings

SPP Set Partitioning Problem

SS Scatter Search

STCP Symmetrical Total Covering Problem

SVRPTWKR Solid Waste Collection Vehicle Routing Problem with time windows, kick-backs and road attributes

SWC Solid Waste Collection

SWCD Solid Waste Collection and Disposal

SWD Solid Waste Disposal

SWM Solid Waste Management

TS Tabu Search

TSP Traveling Salesman Problem

UFLP Uncapacitaed Facility Location Problem

US-EPA United State Environmental Protection Agency

VND Variable Neighbourhood Descent

VNS Variable Neighbourhood Search

VRP Vehicle Routing Problem

VRPTW Vehicle Routing Problem with Time Windows

VRPTWSD Vehicle Routing Problem with Time Windows and Split Delivery

LIST OF SYMBOLS

$\leq$	less than or equal to
$\geq$	greater than or equal to
$\in$	belongs to or element of
$\subset$	subset of
$Co(X)$	the convex hull of X
$\sum$	summation
$d(\cdot,\cdot)$	distance metric
$\Re$	the set of real numbers
$\Re^n$	the $n-$ dimensional real number
$\Re_+$	set of positive real number
Z_+	the set of positive integers
$\lvert\cdot\rvert$	absolute value
$\lVert\cdot\rVert$	euclidean norm
$\lim_{x\to\infty}$	the limit as x approaches ∞
$\max_{x\geq 0}(\cdot)$	maximum value when $x \geq 0$
$\min(\cdot)$	minimum value
x^*	the optimal value of x
$exp(\cdot)$	exponential value
$v(\cdot)$	the value of
$\emptyset$	the empty set
$\exists$	there exists
$\forall$	for all values of
ϵ	a very small arbitrary value

LIST OF APPENDICES

ABSTRACT

Various studies have indicated that the collection phase of solid wastes, which comprises of the initial collection at the source of generation and the transportation to the disposal sites, is by far the most expensive. Two fundamental issues of concern in solid waste collection are the locations of initial collection and the period of collection by the dedicated vehicles. However, considering the prevailing conditions of adhoc location of waste containers and the faulty roads in many developing countries, this research was conducted to develop two effective models for solid waste collection and disposal such that new parameters measuring the capacity of waste flow from each source unit and road accessibility were introduced and incorporated in the mathematical formulations of the models. To formulate the problems, two classes of integer programming problems namely, Facility Location Problem (FLP) and the Vehicle Routing Problem (VRP), were used for the collection and disposal respectively. The clustering process involved in the model for the collection phase was based on the Euclidean distance relationship among the various entities within the study area. In this model, the study area was considered as a universal set and simply partitioned with each element representing a cluster. At this stage, a threshold distance was defined as the maximum allowable distance between a cluster and the potential collection sites. In the VRP formulation of the disposal model, two new parameters, called the ***accessibility ratio*** and ***road attribute***, were introduced and included in the formulation. The inclusion of these parameters ensure that a waste collection vehicle uses only roads with high attributes. The solution to the model on the collection phase was based on the Lagrangian relaxation of the set of constraints where decision variables are linked, while in the model on waste vehicle routing, the assignment constraints were relaxed. Both resulting Lagrangian dual problems were solved using sub-gradient optimization algorithm. It was shown that the resulting Lagrangian dual functions were non-differentiable concave functions and thus the application of the sub-gradient optimization method was justified. By applying these techniques, strong lower bounds on the optimal values of the decision variables were obtained. All model implementations were based on randomly generated data that mimic real-life experience of the study area (Eti-Osa Local Government Area of Lagos State, Nigeria), as well as large-scale standard benchmark data instances in literature. These computational experiments were carried out using the CPLEX and MINOS optimization solvers on AIMMS and AMPL modeling environments. Results from the computational experiments revealed that the models are capable of addressing the challenge of solid waste collection and disposal. For instance, more than 60% reductions were obtained for the number of collection points to be activated and the container allocations for the different wastes considered. Numerical results from the disposal model showed that there is a general reduction in the total distance covered by a vehicle and a slight improvement in the number of customers visited. Result comparison with those found in literature suggested that our models are very efficient.

Keywords: Location, Allocation, Routing, Solid waste collection and disposal, Lagrangian relaxation, Sub-gradient optimization

CHAPTER ONE

INTRODUCTION

This research focused on proposing and solving models associated with the challenges of collection and disposal of solid wastes in low-income developing countries. Of particular interest was the integer programming formulation of the problems and their solutions by the application of Lagrangian relaxation and sub-gradient optimization methods. These approaches involve solving the concave Lagrangian dual functions obtained through the Lagrangian relaxation method within the sub-gradient optimization environment. This research was necessitated by the weight of challenges confronting the collection and haulage of solid wastes in developing countries due to the numbers of existing faulty roads on which waste collection vehicles ply.

1.1 Background

Waste in its solid form is called solid waste. Wahab (2012) described it as wastes generated by households, and wastes of similar nature generated by commercial and industrial layouts, institutions such as schools, hospitals, care homes, and persons, and from public places such as streets, markets, slaughter houses, public toilets, bus stops, parks and gardens. All of these forms of waste may be classified into the following sources: residential, commercial, institutional, industrial, municipal, hazardous and agricultural (Tchobanoglous & Kreith, 2002). According to United State Environmental Protection Agency (US EPA, 2008), municipal solid wastes are materials traditionally managed by municipalities, whether by burning, burying, recycling or composting. This waste is normally assumed to include all the wastes generated in a community, with the exception of waste generated by municipal services, treatment plants and industrial and agricultural processes (Tchobanoglous & Kreith, 2002). The focus of this study is to design optimal approaches for the collection and disposal of this form of waste.

In Coffey and Coad (2010), the phrase 'solid waste collection' was considered to include the initial storage of waste at households, shops or business premises; the loading, unloading and transfer of waste; and all stages of transporting the waste until it reaches its final destination - a treatment plant or disposal site. The sweeping of streets and public places, the cleaning of open storm drains and the removal of these wastes are also included. Apparently, the studies considered the process of transportation (which in the context of this thesis is termed disposal) as being part of collection. Their assertion is not contested, but here the processes of collection and disposal are treated separately. This constitute one of the major

features of this research as no integrated location-routing problem was designed and solved. Rather, separate mathematical models were suggested for the different phases of collection and disposal of solid wastes. Hence, throughout this thesis, the term solid waste collection (SWC) shall mean a process whereby citizens convey their wastes to the nearest collection facilities, whereas solid waste disposal (SWD) shall mean the process of loading waste into waste vehicles at various collection facilities, the transportation of same and unloading at the available disposal sites.

Due to growing world population, Solid waste management (SWM) has become one of the most intractable problems confronting municipal authorities especially in developing countries. SWM comprises three broad activities: collection, transportation and processing. Due to great monetary requirements and environmental impact, the collection and transportation of waste remain the most important aspects of SWM (Coffey & Coad, 2010). SWC involves the provision of services which is usually an agreement between the waste generators and the collection agencies. Collection services available for solid waste depend significantly on two major categories of wastes namely: commingled (unseparated) wastes and source separated wastes. A proper classification of SWC system can be premised on the mode of operation, the types of equipment used or the types of waste to be collected. On the basis of operation mode, the system of collection can be by the following

(a) Haul container system

    (i) Conventional Mode

    (ii) Exchange Mode

(b) Stationary container system

These two types of collection system are illustrated in Figures 1.1-1.3 respectively.

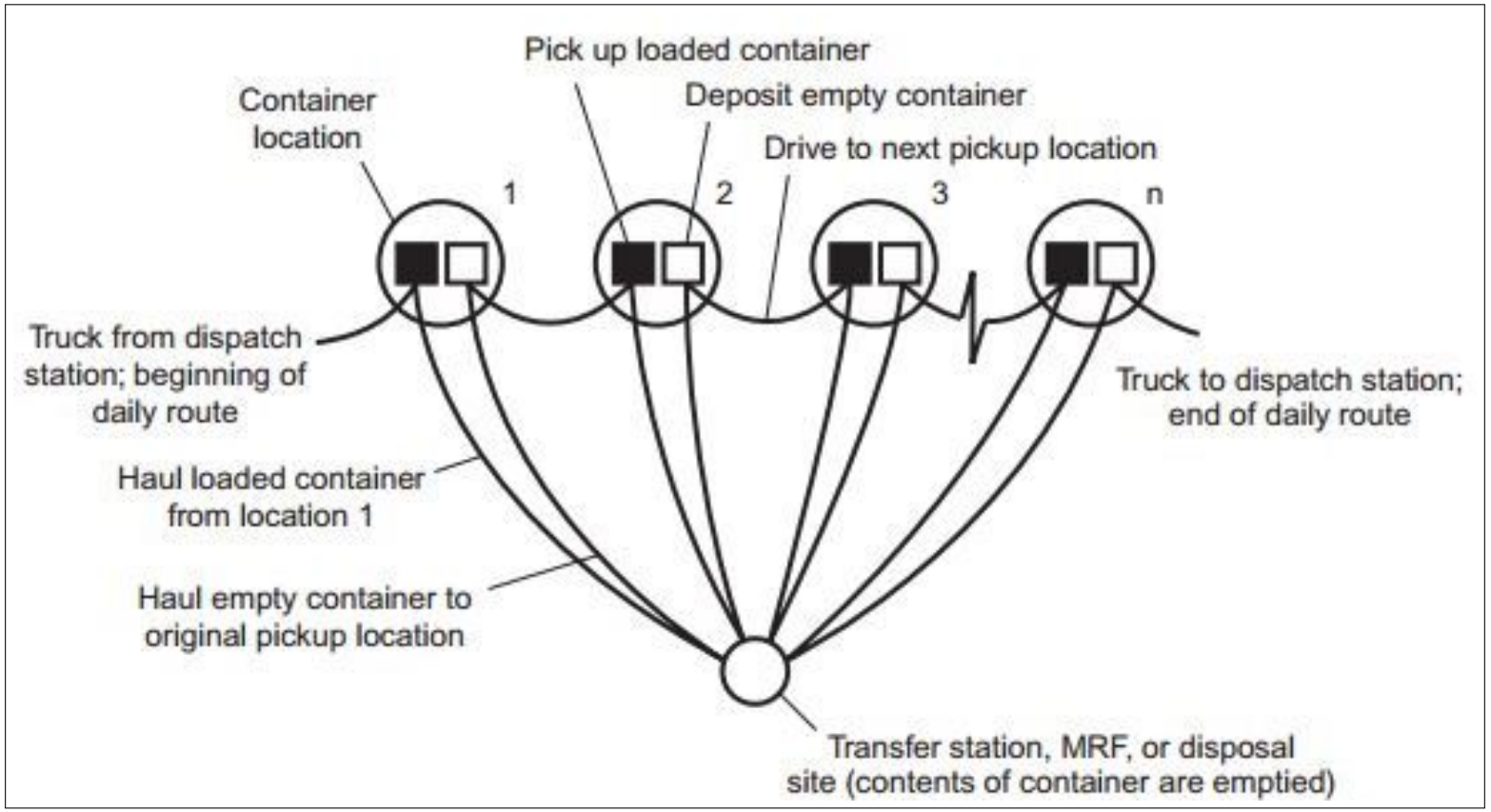

Figure 1.1: Illustration of the Haul Container System (Conventional Mode).
Source: Reprinted from Handbook of Solid Waste Management (page 7.16), by G. Tchobanoglous & F. Kreith, 2002, USA: McGraw-Hill.

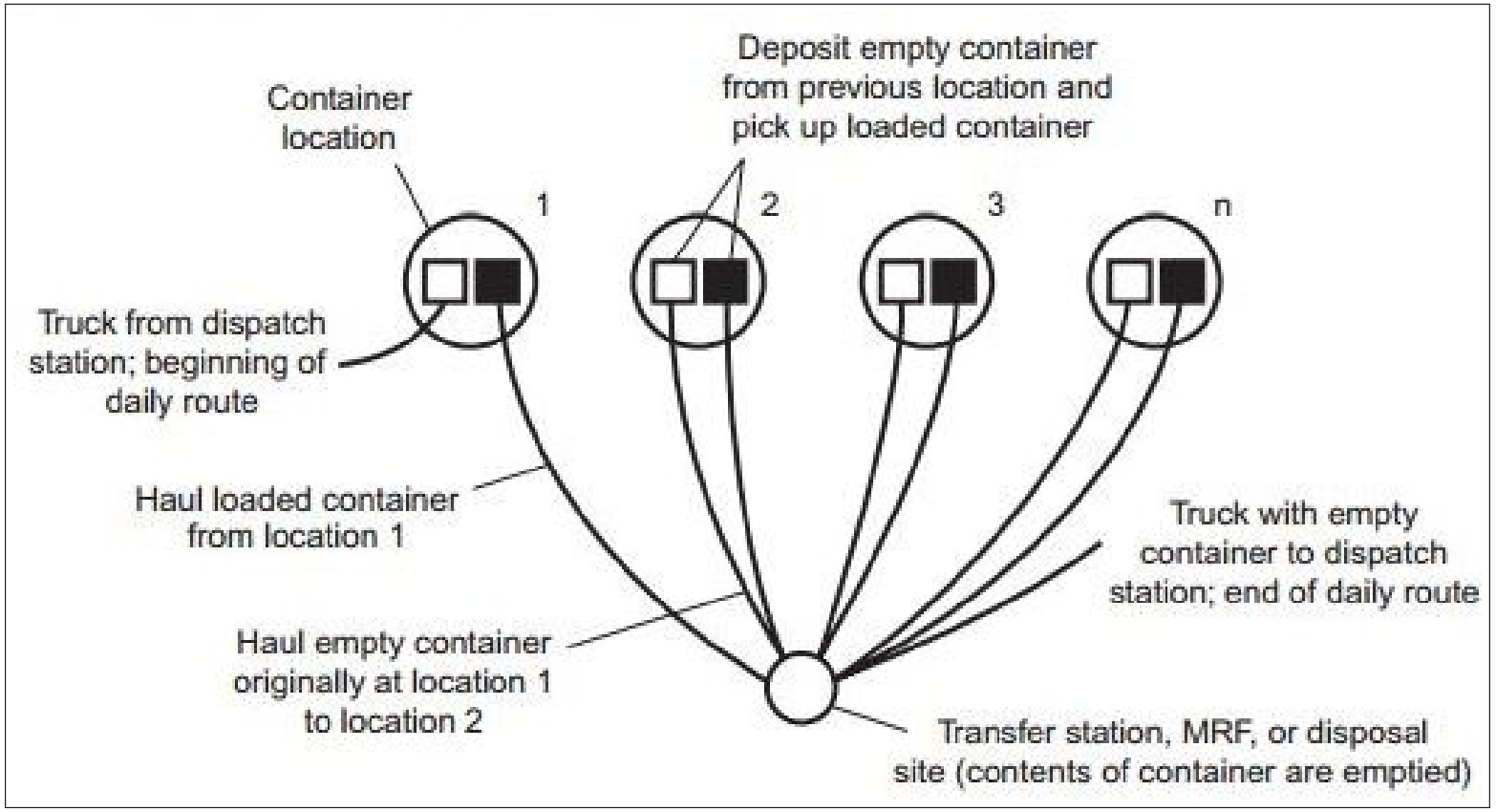

Figure 1.2: Illustration of the Haul Container System (Exchange Mode).
Source: Reprinted from Handbook of Solid Waste Management (page 7.16), by G. Tchobanoglous & F. Kreith, 2002, USA: McGraw-Hill.

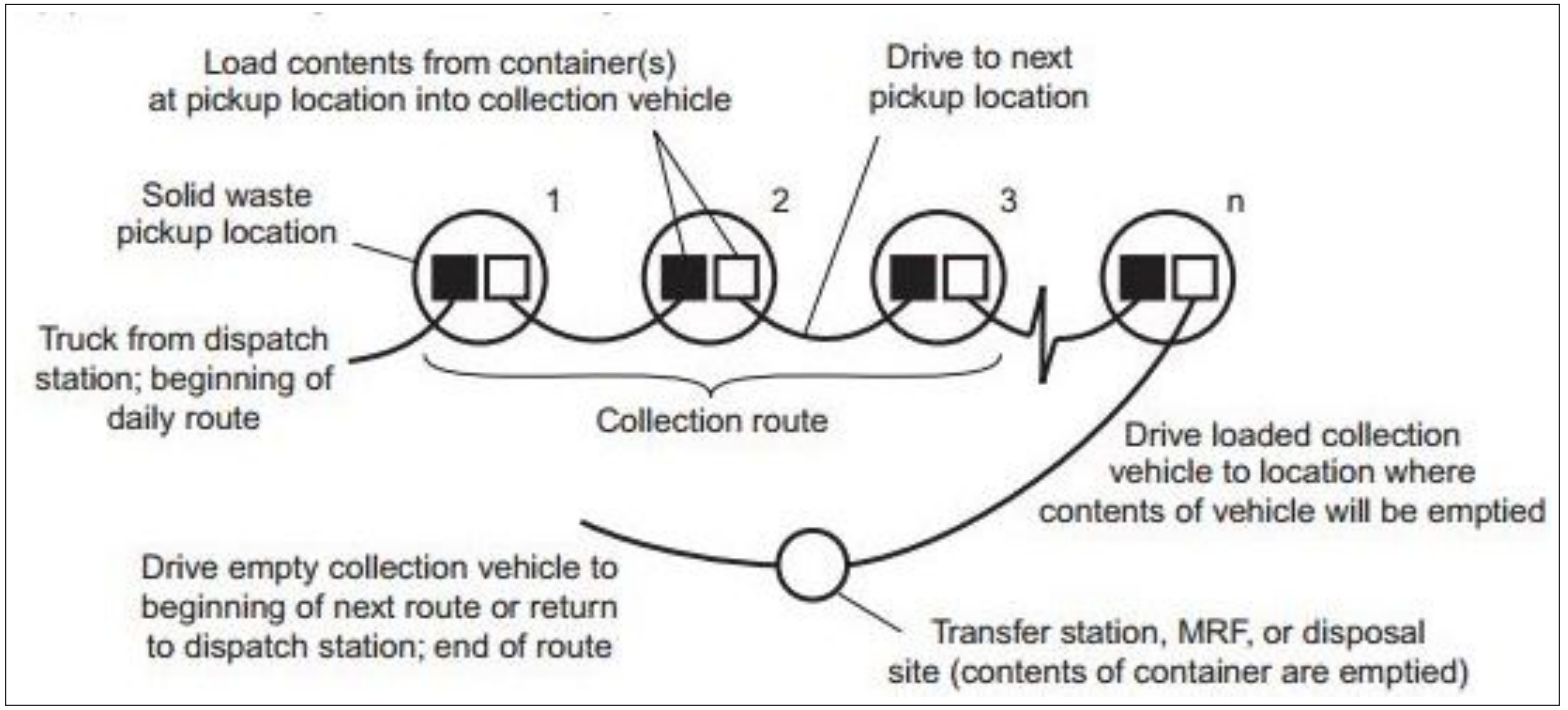

Figure 1.3: Illustration of the Stationary Container System.
Source: Reprinted from Handbook of Solid Waste Management (page 7.16), by G. Tchobanoglous & F. Kreith, 2002, USA: McGraw-Hill.

Under the conventional mode (Figure 1.1), dedicated trucks are used to move loaded containers to transfer station, empty it and then return to their original location. It is ideal for areas with high generation and high flexibility by reason of availability of different sizes and shapes of containers. The exchange mode (Figure 1.2) is characterized by loaded containers used for collection to transport wastes to transfer stations or disposal sites. These containers are emptied and transferred to different location in exchange version. The driver begins the route with empty container from the dispatch stations (depot) to be deposited at the first collection site. This system is advantageous in situations where sizes of the containers are similar. In the stationary container system (Figure 1.3), designated containers remain at the source of generation except when moved to the curb or other location to be emptied. The collection truck is driven from pickup locations to pickup location until it is fully loaded. The choice of system adopted for a particular area depends on several factors that the authorities have evaluated beforehand.

In an effort to design an effective means of collecting solid wastes, it is also important to consider the time of collection and the point of collection. These variables have a great effect on the cost of waste management. Modeling an optimal collection system always revolve around these two variables. When the points of collection are badly designed, it could evidently tell on the overall time of collection and the associated costs. Similarly, when collection is badly scheduled, the after-effect is that most collection stations may not be visited on request. When this occurs, environmental degradation and outbreak of diseases are always the outcomes. Quite a number of studies have worked on the problems of SWCD and a number of them have considered these variables in modeling optimal SWCD systems. For instance, Erkut *et al.* (2008) gave a multi-criteria facility location model for municipal SWM by comparing and contrasting between the regional and prefectural SWM planning in central Macedonia. A multi-criteria mixed integer linear programming (MILP) model to solve the location-allocation problem at the regional level was proposed and a mild non-dominated solution was obtained using the lexicographic minimax approach. Their location-allocation model addresses the transfer stations (collection sites), material recovery facilities (MRFs), incinerators and sanitary landfills with the overall objective of finding the location and topology of those waste facilities.

Badran and El-Haggar (2006) also proposed a model based on MIP to optimize the municipal SWM in Port Said, Egypt. Their objective is to select between the different potential sites for collection stations to minimize the total cost of the entire municipal SWM system. Tralhao *et al.* (2010) considers a similar multi-objective modeling approach for urban sorted waste. However, their model deals with the problem of identifying the locations and capacities of multi-compartment containers. More recently, Ghiani *et al.* (2012) proposed an IP model for capacitated location of waste collection sites in the city of Nardo, in southern Italy. There

are two distinct features of this model: the inclusion of constraints that force each collection site to be capacitated enough to fit the expected waste to be directed to such site and the quality of service requirement that ensures each citizen is served by the collection site closest to him/her, rather than just any close site. Their optimization model minimizes the total number of collection sites to be activated among a set of potential sites. It also determines the optimal allocation of customers to collection sites, as well as allocating containers to these activated sites such that demands are satisfied.

A major problem encountered in the process of waste disposal concerns the waste collection vehicle routing problem. In a typical routing problem, it is assumed that there exists a single depot hosting a homogeneous fleet of vehicles (equal capacity), an operational area (e.g. a city, a residential area, an institution, a hospital, etc.), a set of collection sites and transfer stations or landfills. In some specific instances, each collection site has a time window, that is, the earliest and latest times within which collection service is allowed to take place. The time window may also apply to each vehicle, specifying operational start time and trip completion time. Such problem has received considerable attention in recent years and the results have been applied to real life waste collection processes.

Specifically, Kim *et al.* (2006) focused on a real life waste collection problem of vehicle routing with specification on time windows, multiple trips and drivers lunch breaks. Other related articles have been published on this problem among which are: Ombuki-Berman *et al.* (2007), and Benjamin and Beasley (2010). More recently, Buhrkal *et al.* (2012), addressed the associated problem of specifying time intervals between which customers demand may be satisfied. A mathematical model formulated as a MIP that included drivers lunch and rest breaks was given. To solve the problem, an adaptive large neighborhood search heuristic was proposed to reduce the associated costs of waste disposal. Results were tested on data from the Danish garbage collection company.

From the account above, it is clear that
(i) From the perspective of points of collection, a clustering process that involves individual building was not considered especially as it relates to SWCD. In other words, there seems not to exist a kind of hierarchical clustering that involves an initial grouping of houses, for instance in a residential area, into clusters, which are then regrouped into another set of clusters, this in association with the opened collection sites. The clustering process is based on some distance and capacity constraints.
(ii) The vehicle routing problem is one of the most researched combinatorial problems in the literature and it has been applied by several authors to waste collection. However, no consideration has been given to the choice of route for a vehicle based on the attributes of roads. Due to the deplorable state of most roads, especially in some low-income developing countries, this limitation becomes necessary as the costs of maintenance of heavy waste

trucks are very high and sometimes spare parts are not readily available. Hence, considering the costs of using bad roads, imposing an extra constraint to prevent the use of such will ultimately reduce the cost of waste collection.

Therefore, in this study, these two concerns were handled through a two-stage modeling approach. In the first stage, a model was presented and formulated as capacitated facility location problems (CFLP). This model handled the first issue on clustering. The other model was formulated as a vehicle routing problem (VRP) and handled the additional restriction on road attributes.

The detailed surveys of these two classes of problems are provided in chapter two. Meanwhile, since the formulations of these problems are found in the theory of optimization, effort is made at giving an overview on the subject of optimization, especially as it applies to SWM.

1.1.1 The Concept of Optimization

Optimization is a tool for critical decision-making in any situation where choosing among several alternatives arises. In a broader sense, it involves optimizing (maximizing or minimizing) some function known as the objective function relative to some set of constraints often representing a range of choices available in a certain situation. This objective function allows different choices to be compared in order to determine the most suitable. Common applications in different fields of life may include: cost minimization, profit maximization, optimal design, management and control, best approximation, job scheduling, etc.

Optimization is a very broad discipline which started over a century ago (Sarker & Newton, 2008). It started as a technique in differential calculus for finding the optimal values of differential equations. The earliest application of optimization may be traced to the work of Gantt in 1900 where machine jobs were effectively scheduled through the use of charts. The problems encountered by telephone users were analyzed mathematically by Erlang in 1917 giving rise to the well-known queuing theory of this modern era. As a consequence of World War II, Dantzig proposed one of the strongest techniques of optimization known as the simplex algorithm in 1947 for solving linear programming (LP) problems. This major breakthrough by Dantzig actually established LP problems as one of the most applicable optimization problems. This is especially so as it involves a considerable usage of computer (Freund, 1994). Today, the theory of optimization remains a very active area of research as many varieties of new problems find their solutions through the application of optimization methods.

The use of optimization in addressing a problem entails adequate knowledge of some practical and theoretical fundamentals. These include modeling, analysis and implementation.

Modeling has to do with the setting up of an optimization problem by describing in detail its various aspects including the advantageous and negative features, defining key terms and setting up the mathematical formulation. In other words, an optimization model is a mathematical representation of an optimization problem. Let's consider a general structure of an optimization problem. An optimization problem can be stated as follows (Note that in any optimization process, $Min[f(x)] = -Max[f(x)]$ and the signs of the inequality constraints are reversed).

P1: $\quad Min \quad f(x)$

subject to

$$g_i(x) \leq M, \quad i = 1, \ldots, m \tag{1.1}$$

$$h_j(x) \leq P, \quad j = 1, \ldots, p \tag{1.2}$$

$$x \geq 0 \tag{1.3}$$

where x is known as the decision variable, $f(x)$ is the objective function with single variable x. The number of decision variables (in this case, one) and the number of constraints (m and/or p) may not necessarily be equal. The constraint (1.3) is called the non-negativity constraint and may be necessary in many practical problems. M and P are usually known constants for some class of optimization problems. The above formulation with a single variable may be reformulated for multiple variables as follows: Let $\overline{x} = (x_1, x_2, \ldots, x_n)^T$, then we have

P2: $\quad Min \quad f(\overline{x})$

subject to

$$g_i(\overline{x}) \leq M, \quad i = 1, \ldots, m \tag{1.4}$$

$$h_j(\overline{x}) \leq P, \quad j = 1, \ldots, p \tag{1.5}$$

$$\overline{x} \geq 0 \tag{1.6}$$

The definitions in $P1$ are the same for $P2$ except that x is now an n-dimensional vector variable. In the optimization process, what usually follows is the analysis of the model. At this stage, an expert seeks to exploit all the features of the model by a careful application of some tools of analysis. For instance, a mathematical model formulated as an integer programming problem may be subjected to convex analysis in order to gain insight into

some structures of the problem that may lay good foundation for finding optimal solutions. The objective of doing this is to study the nature of the various parts of the model. This kind of analysis may help to decide accurately the best solution approach to be adopted. Having understood some or all the intricacies in a model through this means, the model is now ready to be fully implemented by subjecting it to appropriate method of solution which could be exact or approximate, depending on the complexity of the problem. Figure 1.4 is a flowchart that depicts the entire processes that lead to effective decision making.

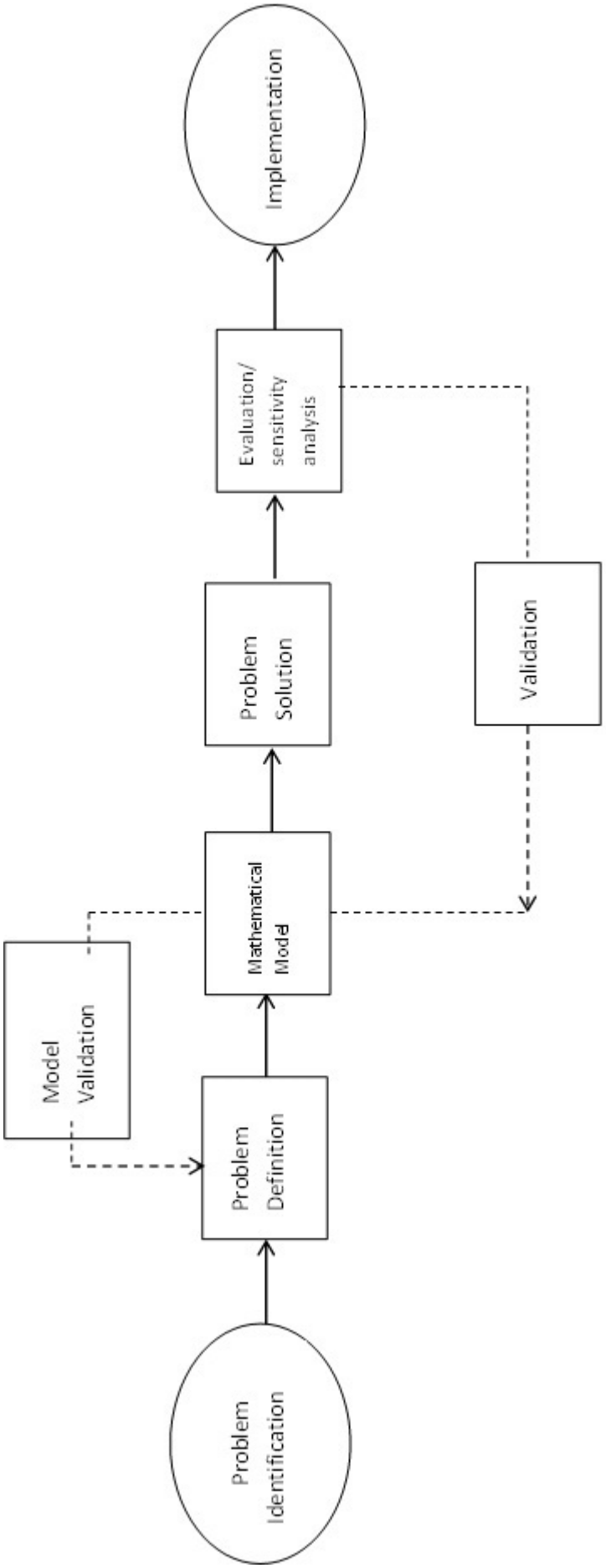

Figure 1.4: Flowchart of Optimization Processes

Prior to finding the solution to a model and the implementation of such solution, tests of validity are usually required because decision makers are often interested in improving the existing system. At the solution stage, this is usually done by the application of sensitivity analysis. With this analysis, the dependency of the optimal solution on the input data is examined. To conduct sensitivity analysis, some alterations are made to the inputted values of some variables at the expense of the previously determined optimal value.

It is evident now that optimization as a mathematical discipline is quite unique. This is because it is furnished with some important features that distinguish it from other disciplines. One of such nice feature is its descriptive nature. Optimization typically describes a situation vividly whenever questions of choice are asked. This form of description allows decision makers to have good idea about the final outcome of their decisions. It is likewise rich in mathematical content and has given rise to a number of interesting disciplines like mathematical optimization, numerical optimization, computational optimization and artificial intelligence (AI).

1.1.2 Classification of Optimization Problems

In classifying an optimization problem, the various features and structures of the problem are examined. Such structures may include, among others, the nature of decision variables, the number of decision variables, the nature of the objective function, the type of constraints, number of objective functions, physical structure of the problem, etc. Problem classification based on the existence of constraints is subject of debate with some mathematicians who held the view that real world situations hardly exist without some limitations (Sarker & Newton, 2008). However, this classification still stands because most constrained problems are solved by transforming them to easier-to-solve unconstrained problems. In Table 1.1, the various classes of optimization problems are presented.

Table 1.1: Classification of Optimization Problems (Adewumi (2015))

s/n	Structure	Characteristics	Classification
1	Number of decision variables	One	Univariate
		More than one	Multi-variate
2	Number of optimal points	One	Unimodal
		More than one	Multi-modal
3	Type of decision variables	Continuous real number	Continuous
		Continuous and integer	Mixed Integer
		Integer in permutation	Combinatorial
4	Existence of constraints	Constraint exists	Constrained
		Constraint does not exist	Unconstrained
5	Nature of objective function	Linear	Linear
		Convex objective and constraint	Convex
		Nonlinear and/or constraint	Nonlinear
6	Nature of decision variables	Probabilistic	Stochastic
7	Physical structure	Controlled, Dynamic	Optimal control

1.1.3 Deterministic Optimization Problems

The classifications based on the type of decision variables in item 3 of Table 1.1 can further be grouped as continuous and deterministic problems. Deterministic problems arise in two forms: continuous and discrete. In a continuous optimization problem, the decision variables are real and assume continuous form. These kinds of problems are common in real life situations. However, there are certain circumstances where optimization problems become absurd when decision variables assume continuous values. In such situations only integer variables (or a mixture of integer and non-integer variables) are considered. For example, in the facility location problem (FLP), a variable x_i could represent the number of hostels of type I that a University authority desires to construct or it could indicate whether or not a particular distance learning centre should be opened in a particular city. The mathematical formulation of such problems includes integrality constraints, which have the form $x_i \in Z^+$, where Z^+ is the set of positive integers; or binary constraints which have the form $x_i \in \{0, 1\}$. This is in addition to some other algebraic constraint(s). Problems satisfying this feature are known as integer programming (IP) problems and are a type of discrete optimization problems. Another widely applicable discrete problem is the mixed integer programming (MIP) problem in which some of the variables in the problem are not restricted to only integer or binary values. In other words some, but not all, variables may assume continuous values. Apart from the values of variables, the underlining feature of discrete optimization problems is that the decision variable x is drawn from a large and finite set. This is sharply in contrast to continuous problems where the feasible set is usually drawn from an uncountably infinite set.

In many instances, continuous problems are quite easier to solve because the smoothness of the functions makes it possible to use information about the objective function and the constraints at a particular point to deduce information about the behavior of the function at all points in the neighborhood of x. Conversely, the behavior of the objective and constraints in a discrete problem may change drastically as move from one feasible point to another is made. Hence, the feasible sets for discrete optimization problems can be assumed to exhibit an extreme form of non-convexity, because the convex combination of two feasible points is, in general, not feasible.

Even though the main concern in this study was on discrete optimization, the importance of continuous optimization techniques cannot be overlooked. This is because most often than not, techniques of continuous optimization play vital roles in solving discrete problems. For instance, the branch-and-bound method for linear IP problems often requires repeated solution of linear programming relaxation in which some variables are fixed at integer values, while the integrality constraints are temporarily relaxed for some other integer variables.

Usually, the resulting sub-problems are solved by the simplex method.

Discrete optimization problems, also known as combinatorial problems, have a wide range of applications. According to Nemhauser and Wolsey (1988), an important widespread area of application occurs in the management and efficient use of scarce resources for maximum production. Under this application is the operational problem of goods distribution, machine sequencing and production scheduling. They also include planning problems such as capital budgeting, facility location, and portfolio analysis. Discrete optimization also finds application in the design of transportation network and automated production systems.

An important and growing area of application of combinatorial optimization occurs in the process of solid waste collection and disposal (SWCD). Because of great importance of the control of disease outbreak within the confine of limited budget, municipal authorities now engage the expertise of optimization analysts in designing systems of optimal SWCD. The design of an optimal SWCD is the main objective of this thesis.

1.2 Statement of the Problem

Two critical issues have been identified in literature as having direct impact on SWCD. These are the identification of the points of initial collection and the periods of final disposal from the collection sites to the disposal sites. These two issues have been the subject of many researches by ways of mathematical models through the use of FLP and VRP. However, a number of factors have been identified as major determinants of the locations of waste collection sites. Some of these factors include accessibility of sites to waste generation sources, quantity of wastes, type of waste, availability of collection equipment and the policies of stakeholders (Coffey & Coad, 2010). Similarly and to a very large extent, the level of road accessibility can have great impact on the efficiency of waste collection vehicles. The challenges faced in the process of collection and disposal of solid wastes may, therefore, be traced to lack of data on the flow rates of waste generated from individual sources and road accessibility and the incorporation of these factors in the SWCD models. Hence, this study was set out to address the challenges associated with the collection and disposal of solid waste by considering the effects of the quantity of waste from individual clusters and accessibility index of the road network in the collection area.

Descriptively, the problem defined an urban area consisting of a number of clusters where wastes were generated. Three different types of waste were considered namely, plastic, paper and food wastes. Thus, there were three types of containers, each for a type of waste. Also, there is a set of potential collection sites where customers can dump their wastes for daily collection. Furthermore, there is a central depot which hosts a fleet of homogeneous multi-

compartment vehicles. A collection site must be visited once in a day. The vehicles tour a set of routes to fulfill their daily capacity requirements before making their stops at the transfer stations or landfills and finally at the depot. The problems to be addressed are stated as follows:
(i) The development of solid waste collection system such that the overall number of open collection sites is reduced.
(ii) The assignment of clusters to activated collection sites chosen from a number of potential sites by defining a set of candidate collection sites and the threshold distance. The threshold distance, say D, is defined as the maximum allowable distance between a customer and a candidate site. Once this distance is violated, such customer (cluster) cannot be assigned to the particular sites.
(iii) The allocation of the different types of containers to the open collection sites such that the capacity of each site is satisfied.
(iv) The routing of vehicles to the collection sites such that the road attribute constraint is satisfied.

1.3 Significance of the Study

The results in this thesis will provide tangible information to waste management decision makers on the best collection and disposal systems to adopt in the face of low monetary budget. The study will, in particular, be beneficial to local authorities in low-income countries where the state of infrastructure is poor. Specifically, it will solve the problems of adhoc location of waste collection facilities and vehicle routing. Furthermore, the approach in this study is very comprehensive that other sectors where strategic and operational decisions on facility location and transportation logistics are made, can also adopt the models developed to their various settings. Moreover, since the results obtained will be published in reputable journal outlets, interested scholars can benefit from the wealth of the novel ideas in this thesis.

1.4 Research Questions

The following research questions form the basis of the results provided in this study.
(i) How do the quantities of solid wastes generated at each particular cluster affect the assignment of clusters to collection locations and the assignment of waste containers to these locations?
(ii) Does the inclusion of new parameter, which measures the accessibility of roads in the collection area, affect the vehicle route construction and the total travel distance covered for

daily waste collection operations?

1.5 Aim and Objectives

The aim of the research is to design an effective urban solid waste collection and disposal system.
The objectives of this research work are:
(i) To formulate new mathematical models capable of addressing challenges relating to SWCD in urban areas.
(ii) To find optimal locations of waste collection sites.
(iii) To perform an optimal assignment of waste generating centers (customers) to the optimized collection sites.
(iv) To perform optimal allocation of waste containers to collection sites.
(v) To minimize the route length covered by waste vehicles when performing disposal duties.

1.6 Research Tools

The various tools used to carry out the findings in this thesis and their underlying theoretical backgrounds are described in this section.

1.6.1 Integer Programming

Integer programming (IP) is a kind of linear programming in which some or all variables are restricted to non-negative integer values. When all the variables take on integer values, then it is called pure integer programming. On the other hand, the mixed integer programming (MIP) model is a variation where some variables are real and some are integers, or at least one variable is an integer. It may have a binary variable, which can be used to identify if any entity is active or not by being assigned 1 or 0, respectively. The assumptions of the models in this research and the mathematical formulations will follow IP. In the following sections, a description of the two classes of problems used in formulating the proposed models in this thesis is presented

1.6.1.1 Facility Location Problem

The facility location problem (FLP) deals with the question of how to select from a given set of potential locations a cost effective subset of locations to place facility(ies). A facility

could represent an electric power plant, hospital, food production plant, a warehouse (depot), petrol station, government office, etc.

The FLP is an important class of problems in logistic management. Facility location and the assignment of entities to such facility usually determine the distribution pattern and the associated characteristics (e.g. time, cost and efficiency) of the distribution pattern. The placement of one or more facilities and the assignment of customers in an optimal fashion does not only improve flow of materials and services offered by the facilities to customers, but also utilizes the facilities in an optimum manner, thus preventing the use of duplicated or redundant facilities (Sule, 2001).

1.6.1.2 Distance Functions in FLP

In the process of finding optimal locations for a set of facilities, an important phenomenon is the pattern of travel distances among the service customers. Ogryczak (2000) observed that most classical FLPs focus on the minimization of the mean distance or the maximum distance to the facilities. Usually, these distances are measured over the set Z^n with their values in Z. However, because Z^n is not a vector space, the notion of distances is often extended and this gave rise to the following general definition:

Definition 1.6.1. *Let $x = (x_1, x_2)$ and $y = (y_1, y_2)$. Then $d(x, y)$ is the distance function between points x and y, with the following properties*

(i) $d(x, y) \geq 0 \qquad \forall \;\; x,y \in \Re$

(ii) $d(x, y) = 0 \leftrightarrow x=y \quad \forall \;\; x,y \in \Re$

(iii) $d(x, y) = d(y, x) \qquad \forall \;\; x,y \in \Re$

(iv) $d(x, y) \leq d(x, z) + d(z, y) \qquad \forall \;\; x,y,z \in \Re$

In the Definition (1.6.1), x and y are defined on the set of real number $\Re$.

In a similar manner, the distance function between the points $x = (x_1, x_2, \ldots, x_n)$ and $x = (y_1, y_2, \ldots, y_n)$ is denoted by $d_{k,p}(x, y)$ called the Minkowski distance of order p, and is defined as

$$d_{k,p}(x, y) = \left(\sum_{i=1}^{n} k_i |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{1.7}$$

When $k_1 = k_2 = \cdots = k_n = k$, equation (1.7) becomes

$$d_{k,p}(x, y) = k \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{1.8}$$

where k is the weight of the distance function $d_{k,p}$ and the equation (1.8) is called the weighted $d_{k,p}$-norm.

However, suppose $k_1 = k_2 = \cdots = k_n = 1$, then (1.7) becomes

$$d_{k,p}(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{1.9}$$

Equation (1.9) is called the $d_{k,p}$-norm and according to experiment, it was shown in Uster and Love (2003) that distances estimated based on the $d_{k,p}$-norm are more accurate than those in the weighted $d_{k,p}$-norm. According to (1.9), some distances have been defined as follows:

Rectilinear distance: This distance describe, for instance, the distance that a vehicle will cover over a square block in a city layout where there exists no one-way routes. It is given by

$$d_{k,1}(x,y) = \sum_{i=1}^{n} |x_i - y_i| \tag{1.10}$$

Because (2.4) is simple to analyze compared to some other forms of distance, it has become popular among researchers (Drezner & Wesolowsky, 2001).

Euclidean distance: This is defined for $p = 2$ as follows

$$d_{k,2}(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^2 \right)^{\frac{1}{2}} \tag{1.11}$$

Equation (1.11) is the metre rule distance because it gives exactly what would be obtained if the distance between two points were measured with a ruler.

The Chebyshev distance: This distance is defined for $p = \infty$ as follows

$$d_{k,\infty}(x,y) = \lim_{p \to \infty} \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} = max\left(|x_1 - y_1|, \ldots, |x_n - y_n| \right) \tag{1.12}$$

By simple inspection, it is obvious that $d_{k,1}$ and $d_{k,\infty}$ assumes discrete values. Thus, from practical viewpoint, the parameter d_2 being continuous is used widely because of its rotation invariance (Farahami & Hekmatfar, 2009).

For each category of the FLP, there is a corresponding suitable distance function. For instance, in the analytic models of FLP, the travel distances follow the rectilinear metric while in network problems consisting of nodes and arcs, distances are measured with respect

to shortest path. The discrete models can usually make use of all kinds of distance functions (Fouard & Malandrin, 2005).

1.6.1.3 Discrete and Continuous Facility Location Problems

FLPs are not uniquely classified in literature. In (Farahami & Hekmatfar, 2009), location problems are divided into four classes: analytic, network, continuous and discrete models. The analytic models are based on simple assumptions such as the fixed costs of locating a facility. They are hardly used to express real world problems. The network models are frequently encountered in transportation planning and other applications that allow tours on routes represented on a network. Mladenovi *et al.* (2007) classified FLP into continuous and discrete problems. In the continuous models, facilities to be located are placed anywhere in the chosen plane and therefore computations are required to determine the best locations with respect to the distances of the demand points (customers' locations). Furthermore, in continuous location models, customers are grouped (using appropriate techniques) and centroid of each group is determined. Each centroid then becomes the *best* location for each group (cluster). This approach of locating facilities is very applicable in highly sensitive strategic planning.

In most practical real cases, estimates in continuous models make discrete models become practicable. Discrete models handle the allocation of customers to a set of potential location points (usually predetermined either from the results of the continuous model or random selection based on past experiences). In other words, the objective in a discrete model is to select from a set of known locations the required number of location for facilities, and then allocate customers to receive service from exactly one of these facilities at minimum cost (Sule, 2001).

Discrete models usually consist of three main components: facilities to be located, a set of locations and the demand points. The facilities have certain features such as total number, type and costs. There are two cases identified for the number of facility. The first is the single facility problem in which only one new facility is to be opened. The more general case is the multi-facility problem where more than one facility are established simultaneously. Facilities in location-allocation problems can also come in different types such as situations where facilities are designed to provide only one or more services. One other major consideration is the satisfaction of demands at these facilities. This in turn gives rise to the variants of uncapacitated and capacitated FLPs. Based on these features, a number of discrete and continuous models have been proposed in the literature for several areas of application. Some of these are described below. Afterward, some solution approaches to FLPs are described with their algorithms.

1.6.1.4 Single Facility Location Problems (SFLP)

The SFLP belongs to the simplest class of location problems. It involves the location of a single new facility in a plane with the aim of minimizing the sum of distances (euclidean or rectilinear) between the proposed new facility and existing (planar) locations. A simple general formulation of the problem is

$$Min \;\; f(\overline{x}) = \sum_{i=1}^{n} w_i d(\overline{x}, p_i) \tag{1.13}$$

where $\overline{x} = (x, y)$ is the distance coordinate of the location of new facility, $d(x_i, y_i)$ is the distance between the new facility and the planar locations, $p_i = (u_i, v_i)$ are the coordinates of the planar locations, w_i represent the weight of existing facilities, i and n are the index and number of existing facilities, respectively.

1.6.1.5 Multi-Facility Location Problem (MFLP)

In MFLP, more than one new facilities are to be optimally located such that each new facility is linked to at least one other new facility. A typical objective function is formulated as follows:

Let N be the set of new facilities to be located with $|N| = n$ and M, the set of existing facilities such that $|M| = m$. Define $w_{ji} \geq 0$ as the weight between each $j \in N$ and $i \in M$ by a unit distance, $v_{jk} \geq 0$ as the weight between each $j, k \in N$ by unit distance, $d(X_j, P_i)$ is the distance between the location of $j \in N$ and $i \in M$, $d(X_j, X_k)$ is the distance between the location of $j, k \in N$ and $P_i = (a_i, b_i)$ is the location coordinates of $i \in M$. The objective function is

$$Min \;\; \left[\sum_{1 \leq j < k \leq n} v_{jk} d(X_j, X_k) + \sum_{j=1}^{n} \sum_{i=1}^{m} w_{ji} d(X_j, X_i) \right] \tag{1.14}$$

The model (1.14) above is a minisum model that finds the locations of new facilities such that the total cost function (sum of costs directly proportional to the distances between the new facilities, and costs directly proportional to the distances between new and existing facilities (Farahani & Hekmatfar (2009)) is minimized.

1.6.1.6 Fixed Costs Capacitated Facility Located Problem (FCCFLP)

In a FCCFLP, the objective is to minimize the fixed costs associated with the potential facilities. The FCCFLP is a *minisum* problem because it seeks to minimize the sum of the

cost of flow between facilities and customers. The fixed cost is a one-time expenditure that varies from location to location which is expected to be recovered during the entire life of the facility. To formulate the problem as a mathematical model, the following sets are defined: C is the set of all n customers indexed with i, F is the set of all candidate facilities indexed with j. The fixed cost for opening j facility is c_j. The transportation cost from facility j to customer i is t_{ij}. α_j is the capacity of facility $j \in F$ and β_i, the demand of customer $i \in C$. The decision variables are

$$x_j = \begin{cases} 1 \text{ if facility } j \text{ is open;} \\ 0 \text{ otherwise.} \end{cases}$$

$$y_{ij} = \begin{cases} 1 \text{ if a customer } i \text{ is assigned to facility } j; \\ 0 \text{ otherwise.} \end{cases}$$

The MIP model is

$$Min \ Z = \sum_{j \in F} \left(\sum_{i \in C} (t_{ij} y_{ij} + c_j x_j) \right) \tag{1.15}$$

subject to

$$\sum_{j \in F} y_{ij} = 1 \ \forall \ i \in C \tag{1.16}$$

$$\sum_{i \in C} \beta_i y_{ij} \leq \alpha_j x_j \ \forall \ j \in F \tag{1.17}$$

$$y_{ij} \leq x_j \ \forall \ i \in C, \ j \in F \tag{1.18}$$

$$x_j \in \{0, 1\} \ \forall \ j \in F \tag{1.19}$$

$$y_{ij} \in \{0, 1\} \ \forall \ i \in C, \ j \in F \tag{1.20}$$

The objective function (1.15) minimizes the total cost (transportation and fixed cost) associated with an open facility. Constraints (1.16) ensure that each customer is allocated to only one facility. Equations (1.17) define the capacity constraints and they ensure that the total demands of customer i assigned to facility j does not exceed the capacity of j. By constraints (1.18), number of open facilities must not exceed the total number of customers in the system. Equations (1.19)-(1.20) are the integer constraints.

1.6.1.7 Capacitated *p*-median Facility Location Problem ($\mathbf{C}_pMFLP$)

This is a discrete problem where, for instance, the list of potential depots is the same as the list of customers. Selected depots are called the medians or concentrators. A p-median

problem is defined as follows. Consider a connected graph consisting of customers with associated network distances from each other, p new facilities are to be opened to satisfy the demands of these customers. A p-median problem optimally locates the p facilities so that the sum of the weighted distances in the network between the customers and their respective closest facility is smallest (Anderson *et al.*, 1998). p-median problems are widely studied because of their relevance to most real life problems.

The CpMFLP is a variant of the capacitated FLP (CFLP)when fixed costs associated to potential facilities are not considered. Using the approach in Lorena and Senne (2004) with a little modification, the problem is described with the following notations. $N = \{1, 2, \ldots, n\}$ is the set of customers to be allocated to potential medians; $x_{ij} = 1$ if customer i is allocated to median j, or 0, otherwise; $y_j = 1$ if median j is selected, or 0, otherwise. Other parameters are P, the number of facilities to be opened, d_i, the demand of customer i, q_j, the capacity of facility j and c_{ij}, the shipment cost from facility j to customer i. The IP formulation of the problem is

$$Min \;\; Z = \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} \tag{1.21}$$

subject to

$$\sum_{i \in N} x_{ij} = 1 \;\; \forall \;\; j \in N \tag{1.22}$$

$$\sum_{j \in N} y_j = P \tag{1.23}$$

$$\sum_{i \in N} d_i x_{ij} \leq q_j y_j \;\; \forall \;\; j \in N \tag{1.24}$$

$$x_{ij} \in \{0, 1\} \;\; \forall \;\; i, j \in N \tag{1.25}$$

$$y_j \in \{0, 1\} \;\; \forall \;\; j \in N \tag{1.26}$$

Constraints (1.22) impose that each customer must be allocated to only one median. Equation (1.23) prevents the total number of opened facility from exceeding the number of facilities required. Total median capacity is ensured by constraints (1.24).

1.6.1.8 Set Covering Location Problems (SCLP)

In a SCLP, customers are allowed to receive services from potential facilities depending on the distances between the customers and the facilities. A customer can receive service from a facility provided the distance between them is equal or less than a predefined value known

as the threshold distance or coverage radius. There are two cases of SCLP depending on the extent of *covering* on the demands from customers. When all the demand points are covered, the problem is called a total SCLP and when only some points are covered, the problem is a partial SCLP. A variant for each class of SCLP is treated as follows.

1.6.1.9 Symmetrical Total Covering Problem (STCP)

This problem was first formulated by Jans and Degraeve (2008), for a lottery problem. Two variables x_j and a_{ij} are defined on S, the set of all demand points, as $x_j = 1$ if facility$j \in S$ covers a point, 0, otherwise; $a_{ij} = 1$ if customer $i \in S$ is covered by facility $j \in S$. The IP formulation is

$$Min \ \ Z = \sum_{j \in S} c_j x_j \tag{1.27}$$

subject to

$$\sum_{j \in S} a_{ij} x_j \geq 1 \ \ \forall \ \ i \in S \tag{1.28}$$

$$x_j \in \{0, 1\} \ \ \forall \ \ j \in S \tag{1.29}$$

$$a_{ij} \in \{0, 1\} \ \ \forall \ \ i, j \in S \tag{1.30}$$

where c_j is the cost associated with a facility j. The objective function minimizes the total cost of covering all the demand points. Constraints (1.28) ensures that each customer is covered by at least one facility.

1.6.1.10 Maximum Covering Location Problem (MCLP)

This problem reported in Berman *et al.* (2003) is a partial SCLP and has the objective of maximizing the total satisfied demands in the network with limited maximum number of facilities. The IP model is

$$Maximize \ \ D = \sum_{i} \sigma_i z_i \tag{1.31}$$

subject to

$$z_i \leq \sum_{j} a_{ij} x_j \ \ \forall \ \ i \tag{1.32}$$

$$\sum_j x_j \leq P \tag{1.33}$$

$$x_j \in \{0,1\} \ \forall \ j \tag{1.34}$$

$$z_i \in \{0,1\} \ \forall \ i \tag{1.35}$$

σ_i is the demand of customer i, the maximum number of limited facility is P. The two decision variables are $x_j = 1$ if a facility is located at j or 0, otherwise; $z_i = 1$ if the demand of customer i is not satisfied and 0, otherwise. a_{ij} is the same as in section (2.2.2.6). Equations (1.32) mean $z_i = 0$ if $\sum_j a_{ij}x_j = 0$ and hence the demand at customer i is satisfied. If $\sum_j a_{ij}x_j = 1$, then the converse occurs.

1.6.1.11 Undesirable Facility Location Problem (UFLP)

The UFLP belongs to a class of FLP known as the *maxisum* models. Unlike the p-median problems where the desirability of facilities allows for the minimization of the objective function relating to distance or cost, in *maxisum* models, the concern is how to locate facilities far from the intended users. For instance, because of the health and environmental implications, locating landfill facilities close to waste collection points may be undesirable.

Daskin (1995) proposed an IP model for the UFLP with the following parameters: P is the number of expected facilities to be opened, σ_i is the demand of customer i, d_{ij} is the distance between customer i and facility j, x_j is the same as in section (2.2.2.6), $y_{ij} = 1$ if the demand of customer i is satisfied by facility j, and 0 otherwise. The mathematical representation is

$$Min \ \ D = \sum_i \sum_j \sigma_i d_{ij} y_{ij} \tag{1.36}$$

subject to

$$\sum_j y_{ij} = 1 \ \forall \ i \tag{1.37}$$

$$\sum_j x_j = p \tag{1.38}$$

$$y_{ij} \leq x_j \ \forall \ i,j \tag{1.39}$$

$$x_j \in \{0,1\} \ \forall \ j \tag{1.40}$$

$$y_{ij} \in \{0,1\} \ \forall \ i,j \tag{1.41}$$

The objective function (1.36) minimizes the weighted sum of demands and distances with

the limitations of equations (1.37)-(1.41).

1.6.1.12 General Formulation Approach in Vehicle Routing Problem

In this section, some basic ideas that govern the formulation of a VRP are summarized. Consider, for instance, a fleet of capacitated vehicle $k \in K$ at a depot designated as node $i = 1$ and a set of customers to be visited, each having a request $j(j = 2, \ldots, n)$. The tours of the vehicles at the depot on the set of arcs, say $A = \{(i, j) : i, j = 1, \ldots, n | i \neq j\}$, such that the total distance covered is minimized while satisfying the requests at each node. Hence, the general objective in a VRP is formulated as

$$Minimize \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ijk}$$

where c_{ij} is the cost of visiting location j from i. This cost could be the distance covered, the time spent on each arc, the amount of fuel consumed, etc. x_{ijk} is a binary variable that takes the value of 1 if vehicle k travels the arc (i, j) and 0, otherwise. To completely prescribe a VRP, various assumptions are usually made resulting in different constraints. It is important to note that the addition of these more complex constraints is the major contributory factor that has given rise to a number of variants of the VRP. To conclude this section, before we present the detailed review, these constraints are introduced and mathematically formulated.

1. Net flow of vehicles: Usually it is assumed that every vehicle leaving a depot must return to the depot at the close of operation. This is captured by the following equation

$$\sum_{i=1}^{n} x_{i1k} = \sum_{j=1}^{n} x_{1jk} \ \forall \ k \in K$$

2. Number of service per customer: Another constraint is that demand at each node must be served exactly once. Hence, a vehicle must enter and exit at a node. This is usually represented as

$$\sum_{i \in N} \sum_{k \in K} x_{ijk} = 1, \ \ \forall \ j \in N; \quad \sum_{j \in N} \sum_{k \in K} x_{ijk} = 1 \ \ \forall \ i \in N$$

   where N is the set of all nodes in the system.

3. Node-vehicle relationship: A common assumption is that the service at each node is performed by one and only one vehicle. This is captured by another binary variable, say y_{ij} which is 1 given any vehicle uses arc (i, j) and 0, otherwise. This variable is

linked to x_{ijk} in the following manner

$$\sum_{k \in K} x_{ijk} = y_{ij} \ \forall \ i, j$$

4. Capacity: Apart from the general attribute of the fleet, the capacity of each vehicle must not be violated, i.e., a vehicle may not carry more than it is designed to carry. Hence, if the capacity of a vehicle is C and d_j is the demand at node $j = 1, \ldots, n$, then the capacity constraint is given as

$$\sum_{i=2}^{n} \sum_{j=1}^{n} d_j x_{1jk} \leq C \ \forall \ k \in K$$

5. Subtour elimination: Subtours are tours that do not start and end at the depot. In the presence of subtours, solutions with cycles using nodes $2, 3, \ldots, n$ are encountered. Since in the desired result, all moves must start and end at the depot, it is important to seek a way of eliminating such cycles that amount to subtours. In Sarker and Newton (2008), a subtour eliminating constraint is defined by

$$\sum_{\substack{(i,j) \in N \times N \\ i \neq j}} y_{ij} \leq |N| - 1 \ \forall \ N \subset \{2, 3, \ldots, n\}$$

y_{ij} retains its meaning above and N is any proper subset of the nodes $2, 3, \ldots, n$ and $|N|$ is the number of nodes in N.

1.6.1.13 Capacitated VRP

A CVRP, according to Laporte (1992), is characterized basically by a single depot and is defined for a set of homogeneous vehicle $k \in K$, an associated service area defined on a graph $G = (N, A)$ such that $N = D \cup C$, where D is the single depot represented at node 0, C is the set of n customers, $i \in C$, A is the set of arc linking the nodes, i.e., $(i, j) \in A$. The cost (distance or time) associated with traveling from node i to j is given as d_{ij} such that $d_{ij} = d_{ji}$ (the case of symmetric CVRP). The decision variable is $x_{ijk} = 1$ if a vehicle k uses the arc (i, j), otherwise $x_{ijk} = 0$. The mathematical IP formulation of the problem is

$$Minimize \ \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} d_{ij} x_{ijk} \tag{1.42}$$

such that

$$\sum_{j\in N}\sum_{k\in K} x_{ijk} = 1 \ \forall \ i \in C \tag{1.43}$$

$$\sum_{i\in N}\sum_{k\in K} x_{ijk} = 1 \ \forall \ j \in C \tag{1.44}$$

$$\sum_{i\in C}\sum_{j\in N} w_i x_{ijk} \leq Q \ \forall \ k \in K \tag{1.45}$$

$$\sum_{i\in N} x_{ijk} - \sum_{i\in N} x_{ijk} = 0 \ \forall \ j \in N, \ k \in K \tag{1.46}$$

$$\sum_{j\in C} x_{0jk} \leq 1 \ \forall \ k \in K \tag{1.47}$$

$$\sum_{i\in C} x_{i0k} \leq 1 \ \forall \ k \in K \tag{1.48}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ijk} \leq |S| - 1 \ \forall \ S \subseteq C, \ |S| \geq 2, \ k \in K \tag{1.49}$$

$$x_{ijk} \in \{0,1\} \ \forall \ i,j \in N, \ k \in K \tag{1.50}$$

The objective (1.42) minimizes the cost associated with the set of routes covered by vehicle $k \in K$. Equations (1.43)-(1.44) imposed that each customer is visited once. The capacity constraint is represented by (1.45), where w_i is the demand at node i and Q is the carrying capacity of each vehicle. Constraints (1.46) balance the inflow and outflow of vehicles. By equations (1.47)-(1.48), each tour starts and ends at the depot. Equations (1.49) are the subtour breaking constraint and (1.50) define the domain of x_{ijk}.

1.6.1.14 VRP with Time Windows

Toth and Vigo (2002) gave the following mathematical formulation of the problem.

$$Minimize \ \ Z = \left[\sum_{k\in K}\sum_{(i,j)\in A} c_{ij}x_{ijk}\right] \tag{1.51}$$

such that

$$\sum_{k\in K}\sum_{j\in\Delta^+(i)} x_{ijk} = 1 \ \forall \ i \in N \tag{1.52}$$

$$\sum_{j\in\Delta^+(0)} x_{0jk} = 1 \ \forall \ k \in K \tag{1.53}$$

$$\sum_{i\in\Delta^-(j)} x_{ijk} - \sum_{i\in\Delta^+(j)} x_{jik} = 0 \ \forall \ j \in N, \ k \in K \tag{1.54}$$

$$\sum_{i\in\Delta^+(j)} x_{i,n+1,k} = 1 \ \forall \ k \in K \tag{1.55}$$

$$x_{ijk}(w_{ik} + s_i + t_{ij} - w_{jk}) \leq 0 \ \forall \ (i,j) \in A, \ k \in K \tag{1.56}$$

$$a_i \sum_{j\in\Delta^+(i)} x_{ijk} \leq w_{ik} \leq b_i \sum_{j\in\Delta^+(i)} x_{ijk} \ \forall \ i \in N, \ k \in K \tag{1.57}$$

$$E \leq w_{ik} \leq L \ \forall \ i \in \{0, n+1\}, \ k \in K \tag{1.58}$$

$$\sum_{i\in N} d_i \sum_{j\in\Delta^+(i)} x_{ijk} \leq Q \ \forall \ k \in K \tag{1.59}$$

$$x_{ijk} \geq 0 \ \forall \ (i,j) \in A, \ k \in K \tag{1.60}$$

$$x_{ijk} \in \{0,1\} \ \forall \ (i,j) \in A, \ k \in K \tag{1.61}$$

The parameters and variables of the model are defined as follows: E and L are the earliest possible start time at the origin and the latest possible arrival time at the origin, respectively. Q is the capacity of vehicle, d_i is the demand at node i. s_i is the service time for customer i, t_{ij} is the travel time for each arc $(i,j) \in A$, w_{ik} is the start time of service at i by vehicle k, and c_{ij} is the cost of traveling from i to j. $[a_0, b_0] = [a_{n+1}, b_{n+1}] = [E, L]$ represents the time window associated with nodes $0, n+1$, $\Delta^+(i)$ are the vertexes that are directly reachable from i, the forward start of i, $\Delta^-(i)$ are the vertexes from which i is directly reachable from i, the backward start from i. The decision variable is $x_{ijk} = 1$ if vehicle k uses arc (i,j), otherwise $x_{ijk} = 0$.

Equation (1.51) minimizes the total travel cost. One exact assignment is made possible by constraints (1.52). The flow of vehicle $k \in K$ is described in constraints (1.53)-(1.55). Constraints (1.56) denote the restriction on service time, while (1.57)-(1.58) are the time windows constraints. Constraints (1.59) ensure that the demands at node i must not exceed the capacity of the vehicle. Nonnegativity and binary conditions are imposed by (1.60) and (1.61) respectively.

1.6.1.15 Periodic VRP

The definition of PVRP is given as follows. Given a complete graph $G = (N \cup A)$ with known costs along the set of arcs A. The set of customers N is given by $N = \{p | p = 0, \ldots, n\}$, where $p = 0$ represents the depot. If we define the route planning period in terms of days, then D is taken as the set of days, indexed with d, i.e., $d \in D$. The following information is important. A scheme is defined as a set of days within the period of planning during which customers are served. Thus assigning a customer to a scheme implies the customer will be

served on everyday of the planned scheme. Let S denote the set of all schemes indexed by $s \in S$. Each scheme is defined by a vector ϑ_{sd} as

$$\vartheta_{sd} = \begin{cases} 1 \text{ if } d \in D \text{ is in the scheme } s \in S; \\ 0 \text{ otherwise.} \end{cases}$$

The following parameters are also defined: c_{ij} is the arc cost $\forall i, j \in A$, where the index $i = 0$ denotes the depot. N_c is the set of customer locations $= N \setminus 0$, Q_i = total demand of each customer i over the planning period, f_i is a fixed number of visits required by each customer i, K is a set of vehicle indexed $k \in K$, C is the capacity of each vehicle $k \in K$. For homogeneous fleet, the capacities are the same.

There are three fundamental decisions that a typical PVRP considers namely: the selection of a scheme from potential scheme for each customer; the assignment of a set of customers to be served by each vehicle on each day; and the routing of vehicles for each day of the route construction period. These decisions give rise to the following definitions and decision variables.

For each customer $i \in N_c$,

$$S_i = \{s \in S : \sum_{d \in D} \vartheta_{sd} = f_i\} \tag{1.62}$$

That is, the PVRP is made to choose a scheme from S_i (where $S_i \subseteq S$) such that equation (1.62) holds.

Note that if

$$|S_i| = \begin{cases} 0, \text{ then there is no feasible solution to the problem } \forall\ i \in N_c; \\ 1, \text{ then each customer } i \in N_c \text{ has only one possible scheme.} \end{cases}$$

The decision variables are:

$$u_{ijk}^{d} = \begin{cases} 1 \text{ if arc } (i,j) \in A \text{ is toured by vehicle } k \text{ on day } d \in D; \\ 0 \text{ otherwise.} \end{cases}$$

$$w_{ik}^{s} = \begin{cases} 1 \text{ if on scheme } s, \text{ customer } i \in N_c \text{ is visited by vehicle } k; \\ 0 \text{ otherwise.} \end{cases}$$

The formulation of PVRP by Christofides and Beasley (1984) as presented by Francis *et al.* (2008) is given here as follows using the notations described above.

$$Minimize \sum_{d \in D} \sum_{(i,j) \in A} \sum_{k \in K} c_{ij} u_{ijk}^{d} \tag{1.63}$$

such that

$$\sum_{s \in S_i} x_i^s = 1, \ \forall \ i \in N_c \tag{1.64}$$

where

$$x_i^s = \sum_{k \in K} w_{ik}^s = \begin{cases} 1 \text{ if customer } i \in N_c \text{ is visited on scheme } s \in S; \\ 0 \text{ otherwise.} \end{cases}$$

$$y_i^d = \sum_{s \in S_i} x_i^s \vartheta_{sd}, \ \forall \ d \in D; \ i \in N_c \tag{1.65}$$

where

$$y_i^d = \begin{cases} 1 \text{ if customer } i \in N_c \text{ is visited on day } d \in D; \\ 0 \text{ otherwise.} \end{cases}$$

$$\sum_{k \in K} u_{ijk}^d \leq \frac{y_i^d + y_j^d}{2}, \ \forall \ d \in D; \ i, j \in N_c (i \neq j) \tag{1.66}$$

$$\sum_{j \in N_c} u_{ijk}^d = \sum_{j \in N_c} u_{jik}^d, \ \forall \ d \in D; \ i \in N; kK \tag{1.67}$$

$$\sum_{k \in K} \sum_{i \in N} u_{ijk}^d = \begin{cases} \mathrm{y}_j^d \text{ if customer } i \in N_c \text{ is visited on day } d \in D; \\ |k| \text{ otherwise. } \forall \ d \in D. \end{cases} \tag{1.68}$$

$$\sum_{i,j \in B} u_{jik}^d \leq |B| - 1, \ \forall \ B \subseteq N_c; \ d \in D; k \in K \tag{1.69}$$

$$\sum_{j \in N_c} u_{0jk}^d \leq 1, \ \forall \ d \in D; \ k \in K \tag{1.70}$$

$$\sum_{i \in N_c} z_i \sum_{j \in N} u_{ijk}^d \leq C, \ \forall \ d \in D; \ k \in K \tag{1.71}$$

where

$$z_i = \frac{Q_i}{f_i}$$

$$\sum_{i,j \in A} c_{ij} u_{ijk}^d \leq R_L, \ \forall \ d \in D; \ k \in K \tag{1.72}$$

where R_L is the maximum route length.

$$x_i^s \in \{0, 1\}, \ \forall \ i \in N_c; \ s \in S_i \tag{1.73}$$

$$u_{ijk}^{d} \in \{0,1\},\ \forall\ (i,j) \in A;\ \ k \in K;\ \ d \in D \tag{1.74}$$

The travel cost is minimized by the objective function (1.63). Equation (1.64) represents the constraints that ensure a feasible scheme is chosen for each customer, while constraints (1.65) define y_i^d on days within the chosen scheme. Constraints (1.66) allow tours only between customers assigned for visit on day $d \in D$. The conservation of flow between arcs is given by equation (1.67) (i.e., the symmetric property is preserved). Equation (1.68) ensure that customers are assigned to routes for days within their scheme. Equation (1.69) eliminates the subtours while (1.70) prevent a vehicle from being used more than once in a day. Constraints (1.71) and (1.72) define the capacities and route lengths respectively. Equations (1.73) and (1.74) are the binary integer variables.

1.6.1.16 VRP with Split Deliveries

The problem is formulated as follows. Let $N = \{1, \ldots, n\}$ represent a set of customers with each pair (i, j), $i, j \in N$ and $i \neq j$ associated with symmetrical travel time t_{ij} and a travelled distance d_{ij}. $q_i (i = 1, \ldots, n)$ denotes the demand of customer i and 0 denotes the depot. it is assumed that a homogeneous fleet at a single depot serves the requests of all customers with each vehicle leaving and returning to the depot. The set of vehicles is $V = \{1, \ldots, m\}$ with capacity C_k, $k \in V$. For vehicle i, the route travelled is given by $R_i = \{r_i(1), \ldots, r_i(n_i)$ where $r_i(j)$ is the index of the j^{th} customer visited and n_i, the number of customers on the route. Since every route ends at the depot, $r_i(n_i + 1) = 0$.

A customer i is assumed to be served within the time interval $[e_i, l_i]$ = [earliest service time of customer i, latest service time of customer i] with $e_i \leq l_i$ and S_i, the service of customer i. For split deliveries, the request of a customer may be fulfilled by more than one vehicle.

The following decision variables are defined

$$x_{ijk} = \begin{cases} 1 \text{ if } j \text{ is visited after } i \text{ by vehicle } k; \\ 0 \text{ otherwise.} \end{cases}$$

y_{ik} = kick-off time of operation at i by vehicle k. $i \in N$, $k = 1, \ldots, m$

z_{ik} = fraction of demand of customer i satisfied by vehicle k, $i \in N$, $k = 1, \ldots, m$

The following mathematical formulation is based on Belfiore *et al.* (2008)

$$Min\ \ (\sum_{i=0}^{n} \sum_{j=1}^{n} \sum_{k=1}^{m} d_{ij} x_{ijk}) \tag{1.75}$$

such that

$$\sum_{j=1}^{n} x_{0jk} = 1, \;\; k = 1, \ldots, m \tag{1.76}$$

$$\sum_{i=0}^{n} x_{ipk} - \sum_{j=0}^{n} x_{pjk} = 0, \;\; p = 0, \ldots, n; \;\; k = 1, \ldots, m \tag{1.77}$$

$$\sum_{k=1}^{m} z_{ik} = 1, \;\; i = 1, \ldots, n \tag{1.78}$$

$$\sum_{i=1}^{n} q_i z_{ik} \leq C_k, \;\; k = 1, \ldots, m \tag{1.79}$$

$$z_{ik} \leq \sum_{j=0}^{n} x_{jik} = 1, \;\; i = 1, \ldots, n; \;\; k = 1, \ldots, m \tag{1.80}$$

$$\sum_{k=1}^{m} \sum_{i=0}^{n} x_{ijk} \geq 1, \;\; j = 0, \ldots, n \tag{1.81}$$

$$y_{ik} + S_i + t_{ij} - M_{ij}(1 - x_{ijk}) \leq y_{jk}, \;\; i = 1, \ldots, n; \;\; j = 1, \ldots, n; \;\; k = 1, \ldots, m \tag{1.82}$$

$$e_i \leq y_{ik} \leq l_i, \;\; i = 1, \ldots, n \tag{1.83}$$

$$z_{ik} \geq 0, \;\; i = 1, \ldots, n; \;\; k = 1, \ldots, m \tag{1.84}$$

$$y_{ik} \geq 0 \;\; i = 1, \ldots, n; \;\; k = 1, \ldots, m \tag{1.85}$$

$$x_{ijk} \in \{0, 1\}, \;\; i = 0, \ldots, n; \;\; j = 0, \ldots, n; \;\; k = 1, \ldots, m \tag{1.86}$$

The objective function represented by equation (1.75) minimizes the total travel distance. Equation (1.76) guarantees that vehicles leave from the depot to the customers. Equation (1.77) ensures each vehicle leaves a customer to arrive back at the depot. By constraint (1.78), the total demand of each customer is fulfilled. Constraint (1.79), prevents vehicle capacity being exceeded. Equation (1.80) ensures that the demand of a customer is only fulfilled if an assigned vehicle tour that route. Combining equations (1.78) and (1.80) produces equation (1.81) which ensures that each customer will be visited atleast once by at least one vehicle. Equation (1.82) sets the service time windows which also removes subtours. Constraint (1.83) enforces that all customers are served within their time windows. Equations (1.84) - (1.85) establish the integrality constraints. Hence, both y_{ik} and z_{ik} are positive variables while x_{ijk} are binary variables.

1.6.2 Lagrangian Relaxation

As noted in Fisher (2004), combinatorial problems exist in two versions. The first is a small number of very easy problems which may be solved in polynomial time of input length. The second is a large class of hard problems for which known solution procedures require exponential time in the worst case. Many hard IP problems, it has been observed, can be viewed as easy problems complicated by a relatively small set of side constraints. One way to reduce this complexity is to dualize these side constraints and an important tool for doing this is the application of the Lagrangian relaxation (LR) approach.

A typical LR takes an IP formulation of a problem and attach Lagrange multipliers to some of the constraints and relax these constraints into the objective function. The resulting problem is then solved to optimality. The solution value obtained gives a lower bound on the optimal solution of the original problem (Beasley, 1993). In many cases, in the application of LR, a number of easier-to-solve sub-problems are obtained, each of which when solved provides a lower bound on the value of the corresponding decision variable. This scenario is encountered in this thesis as the models contain a number of decision variables.

1.6.3 Sub-gradient Optimization

The sub-gradient optimization (SO) is an iterative approach which is usually used to solve Lagrangian dual problems without using a linear programming system. The method, from an initial set of multipliers, generates further Lagrangian multipliers in a systematic fashion. In other words, it is a procedure which attempts to maximize the lower bound value by suitable choice of multipliers.

1.7 Limitation and Scope of the Study

The content of this research is majorly limited in the sense of the data used to implement the models. For some parameters in the models, real life data are not readily available; hence, benchmark test instances which have been custom-designed for various versions of IP were resorted to. The data for implementing the model on the location of collection sites were randomly generated and are related to information about Eti-Osa LGA of Lagos State, Nigeria.

The study reported in this thesis is within the scope of the following areas
(i) Clustering of customers is based on quantity of waste generated by each customer, capacity of the potential collection sites and the distance between the clusters and the potential sites.

(ii) The use of Lagrangian relaxation and sub-gradient optimization to seek solution to the proposed models.
(iii) CPLEX and MINOS solvers were used on AMPL to run the subgradient algorithms.
(iv) Only test instances with 25 and 50 customers of the Solomon (1987) test suite for VRPTW were considered for implementing Model II.
(v) The waste disposal model was limited to the simple case of single vehicle, single depot and a single type of waste.

1.8 Definition of Terms

Definition 1.8.1 (Cluster). *The word cluster has different meaning in different fields of study. Here, the term is used to mean a group of entities (houses or customers) that are closely related in the measure of distance they assumed with respect to some chosen position and a predefined threshold distance.*

Definition 1.8.2 (Disposal). *As mentioned earlier, disposal in this thesis refers to the haulage of waste-by-waste collection vehicle from the open collection sites to the transfer stations or landfills.*

Definition 1.8.3 (Road Attributes). *This term refers to two different choices that a waste vehicle has before it in order to make the next stop. The attributes are termed good and bad depending on a value called the accessibility ratio defined for each route. At some values, the route is assumed good, at other values, it is considered bad. Hence, road attribute defines a set as*

$$R_a = \{good,\ bad\}$$

Definition 1.8.4 (Collection Sites). *These are designated points where waste collection facilities (containers) are positioned for people to dump their garbage before the final disposal at the transfer stations or landfills as the case may be.*

Definition 1.8.5 (Kickbacks). *The term refers to all events that contribute to vehicle stops exclusive of loading at collection sites and unloading at the final disposal sites. Hence, events such as lunch and rest breaks, on-tour vehicle repair, etc., are all instances of kickbacks.*

Definition 1.8.6 (Request/Demand). *Simply put, a full container at any collection site becomes a request or demand that must be satisfied.*

Definition 1.8.7 (Convex Set). *A set $C \subset \boldsymbol{R}^n$ is convex if for $x, y \in C$,*

$$\alpha x + (1-\alpha)y \in C \quad \forall\ \alpha \in [0,1]$$

Thus, the linear combination of any two points in C must yield a point that is entirely in C. If the contrary happens, then C is non-convex (concave).

Definition 1.8.8 (Convex Function)**.** *A function $f : C \to \boldsymbol{R}$ where C is a convex subset of $\boldsymbol{R}^n$ is convex if*

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y), \ \forall \ x, y \in C, \ \alpha \in [0,1]$$

If this condition is not satisfied, then f is a concave function. See the figure below.

Definition 1.8.9 (Convex hull)**.** *The convex hull of a set C, denoted as $Co(C)$, is the intersection of all convex sets containing C, and itself is a convex set. If C consists of a finite number of vectors $c_1, c_2, \ldots, c_n$, its convex hull is*

$$Co(C) = Co(\{c_1, c_2, \ldots, c_n\}) = \{\sum_{i=1}^{n} \alpha_i c_i : \alpha_i \geq 0, \ i = 1, 2, \ldots, n, \ \sum_{i=1}^{n} \alpha_i = 1\}$$

1.9 Organization of the Thesis

This thesis is organized as follows: In chapter one, the background to this research was set out. The basic concept behind the work, which is the theory of optimization was discussed by giving a simple mathematical formulation of an optimization problem. Also covered in this chapter is an introductory study on SWCD and this includes the various modes of collection that exist. Chapter two of the thesis deals with the survey of literature on FLP and VRP. In chapter three, the two methods used for obtaining the results reported in this study were presented. The two newly proposed models for SWCD were also reported in the same chapter. The analysis and implementation of the models are contained in chapter four. The conclusion to this study and a few recommendations are reported in chapter five.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

Research has shown that the collection of garbage entails some important details about the best location of waste collection facilities. This idea has prompted a host of researchers to propose different models or solution approaches to the problem of waste facilities location. The facility location problem does not apply only to waste collection, as it has been applied in almost every area where decision makers are interested in finding optimal locations for their facilities to serve numerous customers. As will be seen later, the approach has also been adopted in this research work to develop a model for finding the minimum locations of waste collection facilities in urban areas.

Another class of problem that has been applied to address a part of the problems of SWM is the vehicle routing problem (VRP). This class of problem is usually adopted to find an optimal set of routes for a fleet of vehicle. The VRP has enjoyed the attention of many researchers, not only in the field of combinatorial optimization, but also in many fields of management science, especially where product distributions are involved. Because of the scores of articles in this area, effort has been made to review quite a number of them.

2.2 Vehicle Routing Problem

The vehicle routing problems (VRPs), also frequently referred to by some as the truck dispatching problems are commonly encountered by organizations with complex operations where it is always of primary interest to find reliable means of obtaining optimal routes to different locations especially as fuel and truck drivers wages (or salaries) are due to increase with time. A very typical example of organizations where VRP is of utmost importance is the newspaper industry where todays product is of no use tomorrow (Golden (1975)).

The VRP was first proposed by Dantzig and Ramzer (1959). It was initially designed with solution algorithm for the problem of delivering gasoline to service stations. A classical VRP, for instance, determines an optimal route for a set of vehicles located at a depot; convening a product to a number of customers at different locations. The majority of problems encountered in real life are usually complicated with the addition of several constraints giving rise to many variants of the VRP. Such constraints include the capacity of vehicle, the time windows of deliveries (or pick-ups), number of depots, number of allowed trips, nature of demands, nature of service and many others. When one or more of these limitations are

introduced with increasing number of customers to be served, the problem turns out to be NP-hard because they cannot be solved in polynomial time. The fact that all VRPs are NP-hard was established by Lenstra and Rinnooy Kan (1981). Solomon and Desrosiers (1988) affirm that VRP with time windows is also NP-hard since it is an extension of the VRP. Similarly, Dror and Trudeau (1990) showed that VRP with split delivery (VRPSD), being a relaxed formulation of the VRP, is also NP-hard. Hence, exact solution techniques for this class of problem may sometimes have to be sacrificed for heuristic methods to obtain near-optimal solution.

2.2.1 Capacitated Vehicle Routing Problem

In a typical CVRP, a vehicle is allowed to visit and serve each customer on a set of routes exactly once. The vehicle starts and ends its visit at the central depot such that the total travel cost (distance or time) is minimized and the vehicle total capacity is not exceeded. This is the basic form of the VRP, because VRP without the capacity constraint and with a single vehicle can also be seen as a Traveling Salesman Problem (TSP) (Mavrovouniotis & Yang, 2015).

A review of some exact algorithms which were proposed in the past few decades and are based on the branch and bound technique can be found in Toth and Vigo (2002). The work contains two separate forms of CVRP namely, the symmetric CVRP (SCVRP) and the asymmetric CVRP (ASCVRP). In the former, the travel cost between two request nodes is the same in both directions, whereas in the latter, the reverse is the case due to imposition of one-way directions in most urban areas. A similar review was carried out by Baldacci *et al.* (2012) and considered an additional constraint of time windows. The authors based their review on problem formulations, relaxation techniques and exact methods for finding solutions. The computational efficiencies of different exact methods were also provided.

According to Lenstra and Rinnooy Kan (1981), the CVRP belongs to the class of NP-hard combinatorial problems. This statement is validated by the volume of literature that considers the use of heuristics to solve the problems. Mazzeo and Loiseau (2004) developed an ant colony algorithm based on the technique presented by Colorni *et al.* (1991). Their algorithm solves problems with approximately 50 nodes. A model with independent route length was formulated using linear integer programming approach by Tarakkoli-Moghaddan *et al.* (2006). The objective of their work was to minimize the cost associated with heterogeneous fleet and maximize the capacity utilization. They proposed a near-neighbourhood-based hybrid SA to solve the problem. In the drive to provide an efficient, effective and practicable means of managing garbage collection system. Tavakkoli-Moghaddan *et al.* (2007) addressed the problem for optimal fleet cost and total travel distance by considering the case of split

services at each demand node and heterogeneous fleet with the aid of mixed integer linear model formulation. They solved the problem by using simulated annealing (SA) technique.

Gendreau *et al.* (2006) introduced the three-dimensional container loading VRP where rectangular (3D) goods containers of fixed size are to be packed into identical rectangular loading compartment of a vehicle. Bortfeldt (2012) extended the work to CVRP and proposes a hybrid algorithm of Tabu search and tree search for optimal vehicle routing and packing plan. The assumption of unlimited supply of vehicles in the tour partitioning heuristic for the VRP proposed by Haimovich *et al.* (1988) was relaxed to the case of fixed homogeneous fleet by Lewis and Sexton (2007). The modification ensures that the heuristic guarantee feasible solution under the assumption of fixed number of vehicles. A major difference between these two techniques is that the latter excludes depot in its initial consideration of the VRP.

The CVRP with stochastic demand, formulated as a set partitioning problem (SPP), was solved by Christiansen and Lysgaard (2007) using an exact branch-and-price algorithm, with the associated sub-problem addressed using dynamic programming technique. Lei *et al.* (2011) extended the problem by adding the time windows constraints. Their model was formulated as a stochastic program with recourse action (when total demand on a route exceed the vehicle capacity)and an adaptive large neighbourhood search (ALNS) heuristic was developed for its solution using modified benchmark instances of Solomon (1987). Mester and Braysy (2007) also developed a two-stage iterative metaheuristic for the CVRP. The method combines the efforts of the guided local search and evolution strategies metaheuristics at the final stage. The initial stage causes a starting solution to be generated using a hybrid cheapest insertion heuristic given by Mester *et al.* (2005). Godinho (2008) also formulated two models to minimize route length and route capacity.

In a distance-constrained CVRP, the objective remains to find vehicle routes that minimize the total travel distance under basic system requirements. Two such problems were proposed by Kek *et al.* (2008) to minimize the number of used vehicle and to design a flexible assignment of start and end depots in a multi-depot system. The authors compared the two models taking Singapore as the case study. Lin *et al.* (2009) proposed a hybrid algorithm of SA and TS. The algorithm combines the advantages of both SA and TS with local search with the idea of not restricting a move by the search algorithm within the solution space which can prove to lower the objective function. The long-run effect is to allow movement in the solution space which results in increasing the value of the objective function.

Several other techniques have been proposed for CVRP. Ai and Kachitvichyanukul (2009) proposed a particle swarm optimization (PSO) algorithm which is a stochastic and adaptive nature-inspired optimization technique (Sharma *et al.*, 2015; Adewumi & Arasomwan, 2016) and a swarm-based method capable of producing low cost, fast, and robust solutions to many

hard optimization problems (Arasomwan & Adewumi, 2014; Olusanya *et al.*, 2015), for finding the solution to a CVRP. Wang and Lu (2009) presented a three-phase hybrid GA to address the CVRP. In the first stage, rather than adopting either of nearest addition method (NAM) or the sweep algorithm (SA), NAM was incorporated into the SA in order to generate a well-structured initial chromosome population. The crossover and mutation probabilities were optimized at the second stage using a response surface methodology (RSM), while the final stage enhances the exploration diversity of the GA by incorporating an improved SA into the GA. Juan *et al.* (2010), using state-of-the-art random number generators, combined a classical heuristic of CVRP with the Monte-Carlo simulation technique to develop a hybrid algorithm called SR-GCWS (Simulated Routing via the Generalized Clarke and Wright Savings heuristic). Liu *et al.* (2010) utilize the combined tool of mathematical programming and graph theory to model a multi-depot CVRP with full truckloads. To solve the problem for minimum empty vehicle movement, they proposed a two-phase greedy algorithm which generates cycles in the first and constructs a closed chain in the second.

Minimizing the sum of arrival time of dispatching vehicles at customers locations gives rise to the cumulative capacitated vehicle routing problem (CCVRP). This problem is a generalization of the traveling salesman problem (TSP) which is NP-hard and included the constraints of vehicle capacity and homogeneous fleet. Ngueveu *et al.* (2010) derive both the lower and upper bounds for this problem from a memetic algorithm using local search properties and the CCVRP properties respectively. Ribeiro and Laporte (2012) address the same problem using an ALNS heuristic on a set of benchmark data instances first proposed by Christofides *et al.* (1979) and Golden *et al.* (1998). A solution-starter-based, two-independent-phased algorithm was developed to tackle the problem by Ke and Feng (2013). The algorithm utilizes different perturbation and local search operators for improving the solution. Lysgaard and Wohlk (2014) proposed a branch-and-cut-and-price technique backed up with computational results. This is probably the first exact algorithm that addresses the CCVRP.

VRP finds an important application in the delivery of relief materials immediately after a major disaster especially in cases of urgency is a critical factor. Rivera *et al.* (2016) presented a new variant of CCVRP with a single vehicle performing multiple trips. The study proposed an exact algorithm for minimizing the arrival time while making comparison with two mathematical formulations of the original problem based on reformulation as a resource constraint shortest path problem. The solution was accelerated by some dominance rules and conditions on the lower bounds. Detailed computational tests on the proposed methods were conducted with problem instance for CVRP as contained in Christofides *et al.* (1979).

Lysgaard (2010) introduced a CVRP which is restricted to pyramidal route. The author describes a pyramidal route as that on which a vehicle first visit customers in ascending order

of index and then in the remaining part of the route, makes a visit to customers in descending order of customer index. An exact branch-and-cut-and-price algorithm was developed such that an exact pricing over elementary routes are attained in pseudo-polynomial time. A hybrid electromagnetism algorithm which mimics the attraction-repulsion mechanism among charged particles was used in Yurtjuran and Einel (2010) to find an improved solution using a relatively new local search technique and iterated swap method with test on several benchmark problems. Szeto *et al.* (2011) proposed an artificial bee colony algorithm which is also a swarm-based heuristic.

Rodriguez and Ruiz (2012) considered the effects of logistic factors such as asymmetric cost, geographical location of depot and customers, demand and minimum vehicle capacity, on solution technique to the CVRP. Their study examined quantitative and qualitative differences between the symmetric and asymmetric CVRP by employing different tools of classical heuristics and metaheuristics such as those of Clarke and Wright (1964) and, Gillet and Miller (1974) among others. Through the use of a purpose-designed optimized crossover operator within a genetic algorithm (GA) framework with the aim of finding an optimal set of delivery routes to give a minimal travel cost, Nazif and Lee (2012) solved a CVRP which was characterized by homogeneous fleet at a single depot. Jin *et al.* (2012) presented a solution using a parallel Tabu search method. Their proposed method makes use of several neighbourhood structures which cooperate via a pool of solution with the goal of exploiting their combined energy. This work was extended in Jin *et al.* (2014) by seeking a solution through a cooperative multiple parallel Tabu search method where the best of solution is exchanged in a common solution cluster.

In order to provide an efficient, effective and practicable means of managing garbage collection system in Indonesia, Kuo *et al.* (2012) presented a change-constrained program (CCP) model formulation of CVRP with fuzzy demand (i.e., fuzzy variables are employed to handle uncertain factors associated with the real application of CVRP). A combination of a hybrid particle swarm optimization (PSO) and GA was used to find an optimal approach to the collection system. The multiple phase neighbourhood search GRASP, a modified version of the GRASP heuristic, was proposed by Marinakis (2012) and uses a stopping criteria that is based on Lagrangian relaxation and sub-gradient optimization. Xiao *et al.* (2012) introduce the concept of fuel consumption rate (FCR) into the CVRP with the objective of minimizing fuel consumption during vehicle tours. The problem was formulated as a mathematical optimization model and a simulated annealing heuristic with hybrid exchange was proposed to solve the problem with instances obtained from Christofides *et al.* (1979) and Golden *et al.* (1998).

Bock and Sanita (2013) reported some approximation results of two variants - the capacitated Orienteering problem in Euclidean graphs and the school bus problem with regret

minimization. The first aim was to maximize profit of collection from nodes whose cumulative demand does not exceed the given vehicle capacity. The other sought to design the best routes for a set of buses with limited capacity to pick and deliver a set of pupils to a school within a minimal regret threshold. Stanojevic *et al.* (2013) proposed a solution approach called the Extended Savings Algorithm (ESA). The idea behind their algorithm was to merge routes and the corresponding formula for computing savings. The ESA, in a dynamical fashion, recalculates savings during the iterative process.

The path-relinking procedure (PRP) is a technique that transforms a current solution into a guiding solution in the least number of moves. That is, one candidate solution is removed from the solution and re-positioned elsewhere within the solution domain. This idea guided the work of Sorenson and Schittekat (2013) in developing a PRP for the capacitated VRP based on distance relocation. The proposed PRP was tested by a separate integration within the Greedy Randomized Adaptive Search Procedure (GRASP) and Variable Neighbourhood Descent (VND) metaheuristics.

More recently, Tlili *et al.* (2014) published a work on this class of problems by imposing a limit on the maximum allowable distance that each vehicle in the homogeneous fleet can travel. The problem was solved for near optimal solution using a hybrid swarm-based metaheuristic that integrates variable neighbourhood search (VNS) within the particle swarm optimization (PSO). Summary of some recent studies on CVRP is reported in Table 2.1.

Table 2.1: Summary of some selected recent publications on CVRP

s/n	Author	Variant	Method of Solution	Source of Data	Real-life Application
1	Lysgaard (2010)	Pyramidal Route	Branch-and-cut-and-price	www.branchandcut.org	Transportation network
2	Xiao *et al.* (2012)	CVRP with FCR	SA	Christofide *et al.* (1979) and Golden *et al.* (1998)	Transportation network
3	Ribeiro and Laporte (2012)	Cumulative	ALNS	Ngueveu *et al.* (2010) and Golden *et al.* (1998)	Job scheduling
4	Bortfeldt (2012)	CVRP with 3-D loading	Tabu and tree search	Gendreau *et al.* (2006)	Transportation network
5	Sorensen and Schittekat (2013)	Classical version of CVRP	Path-Relinking procedure + GRASP + VND	Augerat *et al.* (1995)	Transportation network
6	Jin *et al.* (2014)	Classical CVRP	Cooperative parallel TS	Golden *et al.* (1998)	Transportation network
7	Szeto *et al.* (2011)	Classical CVRP	Artificial bee colony (ABC)	Christofide and Eilon (1969), Christofide *et al.* (1979) and Golden *et al.* (1998)	Transportation network
8	Ai and Kachitvichyanukul (2009)	Classical CVRP	PSO + SR	Christofide *et al.* (1979)	Transportation network
9	Lysgaard and Wohlk (2014)	Cumulative	Branch-and-cut-and-price	www.branchandcut.org	Transportation network
10	Wang and Lu (2009)	Classical CVRP	Hybrid GA		Armed forces transportation route
11	Mehrjerdi and Nadizadeh (2013)	Fuzzy demand CVRP	Greedy clustering Method (GCM)	Random generation	Transportation network
12	Ke and Feng (2013)	Cumulative	Two-phase metaheuristic + local search	Christofide *et al.* (1979), Golden *et al.* (1998) and Ngueveu *et al.* (2010)	Post disaster humanitarian aid distribution
13	Liu *et al.* (2010)	Multi-depot CVRP	Two-phase greedy heuristic + local search	Random generation based on complete Euclidean graphs	Collaborative transportation
14	Stanojevic *et al.* (2013)	Classical CVRP	Enhanced Savings algorithm (ESA)	Christofide and Eilon (1969), Christofide *et al.* (1979) and Augerat *et al.* (1995)	Transportation network
15	Kuo *et al.* (2012)	Fuzzy demand CVRP	Hybrid PSO	Garbage collection system in Palembang City, Indonesia	Garbage collection
16	Lin *et al.* (2009)	Classical CVRP	SA + TS	Christofide *et al.* (1979)	Transportation network
17	Jin *et al.* (2012)	Classical version of CVRP	Multi-neighbourhood parallel tabu search	Golden *et al.* (1998)	Transportation network

2.2.2 Vehicle Routing Problem with Time Windows

In a VRPTW, the classical VRP put into consideration the expected time a particular customer is to be served. The objective is to find, for a fleet of vehicle at a central depot, sets of routes that start and end at the depot at a minimum cost for serving the requests of known customers within a specified time interval. Occasionally, a vehicle may arrive outside the bounds of the time windows, in which case, it either waits or incurs a penalty for the lateness (Koskosidis *et al.*, 1992).

Hashimoto *et al.* (2008) generalized the standard VRPTW by limiting travel times and costs to be time-dependent functions. The resulting problem was addressed using a proposed iterated local search algorithm whose neighborhood consists of little modifications to the 2-opt, cross exchange and Or-opt neighborhoods. Ghannadpour *et al.* (2014) used the assumption of random arrival of real time requests with non-deterministic information for dispatcher on location and request size prior to arrival to formulate a model for the multi-objective dynamic VRP with fuzzy time windows. The authors proposed a GA consisting of three modules: management module to check the state of the system; strategy module to organize information generated by management module; and the optimization module. The model was implemented using the case of hospitals blood bags distribution from one or more central distribution location. An iterative procedure running between two phases called the localized optimization framework (LOF) was introduced by Ursani *et al.* (2011) to develop a localized GA taking the VRPTW as a domain space.

Jiang *et al.* (2014) modeled a new variant of VRP which is limited to heterogeneous fleet and time windows. A two-phase tabu search method was used to solve the problem. The first phase comprises of an extension of the produce in Lau *et al.* (2003) to handle the heterogeneous fleet while the other developed a solution improving procedure. The algorithm was tested with six (6) sets of test cases from literature and another purpose-designed case. A route construction heuristic based on an adaptive parallel scheme was published by Pang (2011). He also proposed a modified nearest neighborhood approach of Solomon (1987) to find the best match of customers and vehicles. In a related work, Cheng and Wang (2009) considered the associated cost of distribution (delivery) centers to model a time-windows VRP based on the work of Solomon (1987). Due to the embedded nature of the problem, the authors decomposed it into two problems: the main and the sub-problem. A GA and a simple heuristic algorithm were developed for the two problems.

A tabu search based on two assignment decision variables were proposed by Nguyen *et al.* (2013) by first assigning vehicles to supply locations followed by the assignment of customers demands to vehicle. The search method was applied to a time-dependent, multi-zone-multi-trip VRPTW. Gutierrez-Jarpa *et al.* (2010) modelled five variants which combine both

delivery and selective pickup at demand locations with a corresponding exact branch-and-price algorithm that minimizes the routing costs in the absence of associated revenue at pickups. A similar algorithm was proposed by Azi *et al.* (2010) for a case where a vehicle is allowed to perform several tours during a workday. A typical example of such application is the transportation of goods with low life span where vehicle routes are short and their combination is necessary to form a full workday. Earlier, Azi *et al.* (2007) had developed a two-phase elementary shortest path-based exact algorithm to solve this kind of problem with computational results reported on Euclidean-type problems.

An enumeration-optimization approach that is capable of addressing medium-sized delivery problems was used by Calvete *et al.* (2007) to solve a goal programming formulated model of VRP with soft time windows. At the enumeration phase, all feasible routes are evaluated, while at the optimization phase, a selection of the subset of the best routes is made from the set of feasible solution. Chiang and Hsu (2014) proposed a knowledge-based evolutionary algorithm (KBEA) to find a set of pareto optimal solution to the multi-objective VRPTW with the objective of simultaneously minimizing the number of vehicles and the total travel distance using the general evolutionary algorithm (EA) approach. Apart from the usual EA parameters of population size and generation count, the study incorporated two more parameters in the work: the exchangeable candidate route in the crossover operator and the number of customer locations to be re-inserted in the mutation operator.

The scatter search (SS) heuristic is an evolutionary technique (Marti *et al.*, 2006) that operates on a set of reference solutions to generate new solutions through weighted linear combinations of structured subsets of solutions (Belfiore & Yoshizaki, 2013). The first description of the method was given by Glover (1997). Other authors have worked extensively on the technique such as Alegre *et al.* (2007), Yamashita *et al.* (2006), Marti *et al.* (2006). Recently, Belfiore and Yoshizaki (2013) proposed a SS technique to solve a VRPTW with fleet size and mixed vehicle constraints with the objective of serving customers within a framed time window using heterogeneous fleet not exceeding minimal costs. Vidal *et al.* (2012) used a hybrid genetic search metaheuristic with three components of assignment, sequencing and route evaluation.

Challenges associated with the distribution of perishable goods, Azi *et al.* (2007) proposed an exact algorithm based on the elementary shortest path technique to solve the single vehicle multi-trip routing problem with time windows in two phases. In the first phase, all non-dominant routes are generated while in the second phase route selection and sequencing are done to form the daily workload for the vehicle. This work was extended in Azi *et al.* (2010) to the use of multiple vehicles and solving the problem with a branch-and-price method where the lower bound value on the minimum total distance was computed by solving the linear programming relaxation from the column generation sub-problems. These

two works were tested numerically using the standard benchmark test instances of Solomon (1987). Other algorithms have also been proposed such as memetic algorithm of Nagata *et al.* (2010). A summary is found in Table 2.2.

Table 2.2: Summary of some selected recent publications on VRPTW

s/n	Reference	Variant	Model formulation approach	Method of solution	Source of test problem
1	Le Bouthillier and Crainic (2005)	Single-depot VRPTW	Formal description	Parallel cooperative multi-search	Solomon (1987), Homberger and Gehring (1999)
2	Mester and Braysy (2005)	Classical VRPTW	Formal description	Active guided evolution techniques	Christofides *et al.* (1979), Taillard (1993,1995), Fisher (1994), Russell (1995), Gehring and Homberger (1999)
3	Homberger and Gehring (2005)	VRPTW with a central depot	Formal description	(μ, λ)-evolution strategy + Tabu search	Solomon (1987), Homberger and Gehring (1999)
4	Azi *et al.* (2007)	VRPTW with single vehicle and multiple routes	IP	Elementary shortest path-based algorithm	Solomon (1987)
5	Qureshi *et al.* (2009)	VRP with soft time windows	MIP	Column generation	Solomon (1987)
6	Nagata and Braysy (2009)	Classical VRPTW	Formal description	Route elimination algorithm	Gehring and Homberger (1999)
7	Ghoseiri and Ghannadpour (2010)	Multi-objective VRPTW	Goal programming	Multi-objective genetic search	Solomon (1987)
8	Ursani *et al.* (2011)	Classical VRPTW	Formal description	Localized optimization framework	Solomon (1987)
9	Balseiro *et al.* (2011)	Time-dependent VRPTW	Integer linear programming (ILP)	ACO + Insertion heuristic	Solomon (1987)
10	Jiang et al, (2014)	VRPTW with heterogeneous fleet	MIP	A two-phase tabu search	Golden *et al.* (1984), Solomon (1987), Liu and Shen (1999), Choi and Tcha (2007)
11	Kumar *et al.* (2014)	Multi-objective VRPTW	Goal programming	Fitness Aggregate Genetic Algorithm	Solomon (1987)
12	Dhahri *et al.* (2014)	Classical VRPTW	ILP	VNS	Solomon (1987)

2.2.3 Periodic Vehicle Routing Problem

The periodic vehicle routing problem (PVRP) generalizes the classical VRP and it involves the construction of vehicle routes over a period of time. This class of problems arises in many real world events including waste collection, courier services, machine replenishment, distribution of materials, and many more. In a typical PVRP, during each time spanning over the entire period of route construction, a fleet of capacitated vehicles tour routes that begin and end at a single depot. The goal of a PVRP is to find a set of routes for each utilized vehicle that minimizes the total travel cost (e.g. distance) such that basic requirements of customers' demands and vehicle capacities are satisfied.

The complexity of the problem prompted Christofides and Beasley (1984) to propose the use of a medium relaxation method to approximate the cost of a PVRP as the sum of radical distances between customers and the depot.

Other formulations of the problem exist in literature. Among these is the work of Tan and Beasley (1984) which simplified the VRP formulation by Fisher and Jaikumer (1981). Their formulation aims to eliminate computational complexity by not stating explicitly the routing constraint. They removed the allocation of customers to vehicle in order to reduce the size of the problem. This formulation is an extension of the general assignment problem with addition of route components.

There are few variants of PVRP, all of which are similar to the VRP models with single visit. Cordeau *et al.* (1997) presented the multi-depot PVRP (MDPVRP) by associating depots with day in the planning procedure. Hadjiconstantinou and Baldacci (1998) use a heuristic based on a similar ordering rule of Christofides and Beasley (1984) to solve a PVRP that combines periodic properties and multi-depot constraints. This method assigns customers to depots thereby defining the service restrictions for each depot. A very similar problem was proposed by Angelelli and Speranza (2002), called the PVRP with intermediate facilities (PVRPIF). This variant does not make use of multi-depots, instead it introduces the idea of "roll-off" points where vehicles can make their stop-overs along their routes thereby giving room to capacity replenishment. They used tabu search to solve the resulting problem.

Another variant is the PVRP with time window (PVRPTW). This additional constraint makes the problem more complex. The problem involves constructing different vehicle routes that ensures all customers are visited within a specified time window in order to satisfy their desired service requests. Cordeau *et al.* (2001) proposed a tabu search heuristic for this problem, a modification of the one contained in Cordeau *et al.* (1997). The authors also constructed new data instances for the PVRPTW and MDPVRP with time window. Francis *et al.* (2008) gave a further extension of the PVRP characterized with service frequency and

vehicle capacity requirement called the PVRP with service choice (PVRP-SC). The authors combined Lagrangian relaxation and branch-and-cut methods to solve the problem. Their result shows that savings depends largely on the geographical location of customers.

The PVRPs have been used in solving quite a number of real-world problems. Foremost among these is described in Banerjea-Brodeur *et al.* (1998). They implemented the PVRP to plan the deliveries of linen materials to 58 different clinics within a healthcare centre using the tabu search technique of Cordeau *et al.* (1997).The application in planning routing and scheduling of service team in an elevator maintenance project was conducted by Blakely *et al.* (2003) and was solved using a multi-phase algorithm which included: a clustering phase; a solution-improving phase guided by a tabu search; and a route-formation phase.

Banking of blood products is an important component of good hospital practices. Hemmelmayr *et al.* (2009), placing significant importance on consistency in deliveries, investigate the application of PVRP to the periodic delivery of blood products to hospitals by the Red Cross Society of Austria. Their problem was modeled as an integer program as well as a PVRP, with solution provided using an IP solver and a variable neighbourhood search method respectively.

Beltrami and Bodin (1997) modeled the problem of waste collection and recycling as a PVRP. Pontin *et al.* (2004) formulated the problem of moving infectious wastes from several facilities to a single disposal site. They proposed a genetic algorithm to obtain a near optimal solution. Teixeira *et al.* (2004) consider the application of PVRP to waste collection with a special case of multiple waste types (three different types of waste) using a cluster-first-route-second heuristic. Table 2.3 displays a collection of other works on PVRP.

Table 2.3: Summary of some recent publications on PVRP

s/n	Reference	Variant	Model Formulation Approach	Method of solution	Data usage/ area of application
1	Francis and Smilowitz (2006)	PVRP with service choice	Continuous approximation	Geographic decomposition & variable substitution	Benchmark test instances
2	Hemmelmayr *et al.* (2009)	PVRP with time windows	Problem description	VNS	Benchmark test instances
3	Vidal *et al.* (2010)	Multi-depot PVRP	Problem description	Hybrid GA	Benchmark test instances
4	Nguyen *et al.* (2011)	PVRP with time windows	MIP	Hybrid GA	Benchmark test instances
5	Yu and Yang (2011)	PVRP with time windows	Problem description	Improved ACO	Benchmark test instances
6	Hamzadayi *et al.* (2013)	Periodic travelling salesman problem (PTSP)	IP	Nested SA	Retail distribution system
7	Zhang et al. (2013)	Multi-period VRP with profit	Problem description	Memetic algorithm	Modified benchmark problems
8	Michallet *et al.* (2014)	PVRP with time spread constraints on service	MIP	Multi-start iterated local search	Shipment transportation
9	Dayarian *et al.* (2015)	Multi-period VRP	2-stage a priori optimization design	Branch-and-price algorithm	Benchmark problems/ product distribution
10	Archetti *et al.* (2015)	Multi-period VRP with due date	Mathematical programming with valid inequalities	Branch-and-cut algorithm	Benchmark problems/ city distribution
11	Luo *et al.* (2015)	Multi-period VRPTW and limited visiting quota	MIP	3-Stage solution via: Decomposition, repair and distance resolution	Benchmark test instances

2.2.4 Vehicle Routing Problem with Split Delivery

The mathematical formulation and economic relevance of the vehicle routing problem with split delivery and time windows constraints was first presented in literature by Dror and Trudeau (1989, 1990). Dror *et al.* (1994) built on this and proposed an integer programming model of the VRPSD, building several classes of valid inequalities. A branch-and-bound algorithm with relaxed constraints was proposed to solve the problem. Two works were published by Frizzell and Giffin (1992, 1995) where both construction and improvement algorithms were developed for the VRPSD. The time window constraints were considered in the later work.

A polyhedral study of the problem was conducted by Belenguer *et al.* (2000). A cutting plane algorithm for small instances was developed to propose a lower bound to the problem while a branch-and-bound algorithm was implemented to obtain integer values. A worst-case performance analysis showing a maximum 50 percent cost savings when split delivery is allowed in a VRP was carried out by Archetti *et al.* (2006b). Earlier, the same authors implemented a tabu search algorithm for the VRPSD in such a manner that at each iteration a neighbour solution is obtained by removing a customer from the current route, to be re-inserted in a new or existing route having enough capacity. A two-phase tabu search heuristic was proposed for the VRPSD by Ho and Haugland (2004). The first phase (a solution constructing phase) uses node interchange while the second, the improvement phase, introduces and eliminates splits. Belfiore and Yoshizaki (2006) uses a scatter search (SS) to solve a VRPTWSD characterized with heterogeneous fleet.

2.2.5 Dynamic Vehicle Routing Problem

In a dynamic VRP (DVRP), unlike other variants of VRP, some part or all necessary information for planning vehicle routes are only known after the commencement of the execution of the original route design. In this manner the static nature of the constructed route becomes dynamic. Usually in a DVRP, a vehicle serves two different types of request: the static (known) requests and the dynamic (unknown/real time) requests. This in turn gives rise to a situation of either partial or full re-planning of the vehicle routes. Inserting real-time customers is usually difficult due to the complexity of the original VRP. Extra constraints consequently are difficult to insert forcing dynamic requests to be ignored in most cases (Larsen *et al.*, 2008). Pictorially, using the case of solid waste collection within an urban setting, a DVRP is depicted as follows:

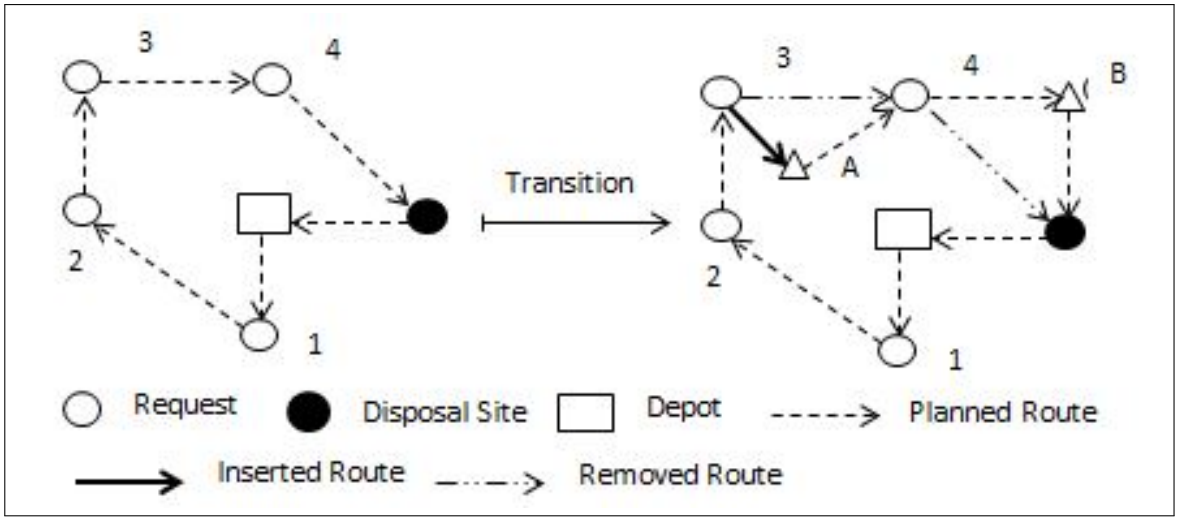

Figure 2.1: (a) Route plan without dynamic information (b) Re-planned route with dynamic information

In Figure 2.1(a), the original route plan involves a vehicle leaving the depot at the start of operation and serving the requests at nodes 1, 2, 3, and 4 in that order before unloading at the disposal site and then returning to the depot. Hence, activities start and end at the depot without alteration to the original route design. In this way the sequence of operation is (Depot-1-2-3-4-Disposal site-Depot). On the other hand, in Figure 2.1(b), new service requests are received at nodes A and B after the start of execution of the initial plan. These dynamic requests are fed to the driver using appropriate technology. The driver in turn waits to receive an updated plan that minimizes the cost of collection and disposal. According to Figure 2.1(b), the sequence of execution of the updated plan is now (Depot-1-2-3-A-4-B-Disposal site-Depot).

In literature there are two broad classes of DVRP namely the deterministic and the stochastic DVRP. In the deterministic version, some or the entire request is unknown and are only received during the execution of the route plan. The stochastic version, on the other hand, exploits the stochastic features of the newly received information during route execution (Pillac *et al.*, 2011). Also, there are two general optimization methods available for inserting dynamic information in VRP. The apriori optimization methods use stochastic or probabilistic information on unknown future events (i.e., the source of dynamism) to design routes. In this way routes are constructed prior to the commencement of operation. The real-time optimization methods construct the route to be executed at the middle of the operation (Larsen, 2001).

There are a number of factors that are capable of altering a static route design. However customers requests remain the most common source of dynamism when implementing a vehicle route. These requests can be for goods (Hvattum *et al.* (2007), Goel and Gruhn (2008)) or services (Thomas, 2007). Other sources studied in literature include travel time (Lorini *et al.* (2011), Guner *et al.* (2012)) and vehicle breakdown (Li *et al.* (2009a), Mu *et al.* (2011)). The degree of dynamism in a VRP can be characterized by either the rate of change of requests or the urgency of requests. The rate of change defines the frequency at which new information becomes available. The urgency of request refers to the length of time between the arrival of a new request and its expected service time. Authors have used these features to measure the level of dynamism existing in route planning. Lund (1996) defined δ, the degree of dynamism, as

$$\delta = \frac{d_r}{t_r}$$

i.e., the ratio between the number of dynamic requests, d_r, and the total number of request

t_r. Larsen (2001) proposed the effective degree of dynamism as

$$\delta^e = \frac{1}{t_r} \sum_{i \in R} \frac{d_{it}}{T}$$

where T is the length of the planning period, R is the set of requests, d_{it} is the arrival time of request $i \in R$ and t_r is as defined earlier. He also extended δ^e to cover problems with time windows reflecting the urgency of requests. He defined the extended δ^e as

$$\delta^e_{ext} = \frac{1}{t_r} \sum_{i \in R} \left(1 - \frac{e_{it} - d_{it}}{T}\right)$$

with e_{it} as the end of corresponding time windows.

Wohlgemuth *et al.* (2012) modified δ^e_{ext} in their study of relief planning in the event of disaster. The modification includes the requirement of capacity of demand and travel time and is given by

$$\delta^e_{ext \times tt \times cap} = \frac{1}{t_r} \sum_{i \in R} \left(1 - \frac{e_{it} - d_{it}}{T}\right) \times \left(1 + \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\frac{E_{ij} - A_{ij}}{A_{ij}}\right)\right) \times \left(1 + \frac{p_{uv} - a_{uv}}{a_{uv}}\right)$$

where E_{ij} is the actual extension of travel time between request nodes i and j, A_{ij} is the assumed travel time between nodes i and j, a_{uv} and p_{uv} are the average and positive vehicle utilization parameters. Hong (2012) expressed δ in term of percentage in the following way

$$\delta = \frac{N_{dr}}{N_{dr} + N_{sr}} \times 100$$

N_{dr} is the number of dynamic requests and N_{sr} is the number of static requests.

Among the earliest works on DVRP are Wilson and Colvin (1977) and Psaraftis (1980). The reader may consult these two materials on the mathematical formulation of the problem. In the few paragraphs that follow, some of the most recent researches on DVRP are reviewed and summarized in Table 2.6.

The past three decades have seen different studies approached the problem from different perspectives, majority of which are based on objectives, source of dynamism, methods of solution and addition of constraints. These basic features or their combinations have given rise to several variants in literature. The addition of time windows constraints, for instance, gave rise to the DVRP with time windows (DVRPTW). These constraints allow the service requests of customers to be met at a particular time of the day. For this class of problem, Qureshi *et al.* (2010) used the well-known Dantzig-Wolfe decomposition (column generation) method to reduce the cost and lateness incurred by freight carriers due to dynamic travel time occurring at the middle of operation. The authors employed a simulation software known as

VISSIM to simulate the traffic network under both normal and congestion conditions. The case considered in their work allows late deliveries (i.e., soft-time windows). A case where lateness is not allowed (hard-time windows) was considered by Hong (2012). He proposed a LNS to solve the problem after an initial decomposition into a series of static VRPTW. A penalty cost approach was used by Ferrucci and Bock (2015) to control route diversion whenever new requests are received and updated in the route plan.

A knowledge-based modelling approach that integrates policies for handling disruptions during the execution of route plan, local search algorithm and object-oriented modelling was proposed by Hu *et al.* (2013). The method was designed to handle unexpected events arising in the distribution processes of urban companies. Queuing theory was used by Van Woensel *et al.* (2008) to capture travel time in an aprior manner. Their solution to the problem was based on local search algorithm which generated solutions for three different road types. Ferrucci *et al.* (2013) inserted stochastic requests using tabu search for solution.

Another variant is the multi-period DVRP (MPDVRP). This is also an extension of PVRP where more than one operational periods are constructed to service customers requests. In Albareda-Sambola *et al.* (2014), service requests with known probability distributions were modelled into the system using adaptive service policy. This technique uses compatibility indices between pairs of customers. These indices are based on geographical distributions. Wen *et al.* (2010) modelled this problem as a mixed integer linear program and proposed a three-phase heuristic. This technique works over a planning period such that the total travel costs with customer waiting time are minimized. The first phase makes a selection of customers to be served within a given planning period. In the second phase, routes are constructed as a PVRP for the selected customers with service frequency equal to 1. Tabu search algorithm was used in the final phase to re-optimize routes to be executed on any day of the planning period. Table 2.4 highlighted some other relevant studies on DVRP.

Table 2.4: Summary of some selected recent publications on DVRP

s/n	Author	Variant	Source of dynamism	Solution Approach
1	Haghani and Jung (2005)	Time-dependent DVRP	Service requests & Travel time	Genetic algorithm
2	Fabri and Recht (2006)	DVRP with pickup and delivery	Vehicle waiting time	Local search
3	Angelelli *et al.* (2009)	Dynamic Multi-period VRP	Service request	Competitive analysis
4	Van Woensel *et al.* (2008)	General DVRP	Travel time	Local search heuristic
5	Branchini *et al.* (2009)	DVRP with stochastic requests	Service request	Construction and an adaptive granular local search method
6	Wen *et al.* (2010)	Dynamic Multi-period VRP	Customers orders & service period	A three-phase rolling horizon heuristic
7	Lorini *et al.* (2011)	General DVRP	Customers requests & Travel time	Extended solution approach of Potvin *et al.* (2006)
8	Liao and Hu (2011)	General DVRP	Travel time	Sweep and TS algorithms
9	Qureshi *et al.* (2012)	DVRP with soft time windows	Travel time	Column generation and micro-simulation
10	Hong (2012)	DVRP with hard time windows	Customers requests	Large neighbourhood search
11	Pillac *et al.* (2012)	DVRP with stochastic demand	Customers demand	Adaptive VNS
12	Wohlgemuth *et al.* (2012)	DVRP with pickup and delivery	Travel time and unknown orders	Tabu search
13	Ferrucci *et al.* (2013)	General DVRP	Stochastic requests	Tabu search
14	Hu *et al.* (2013)	General DVRP	Demand and vehicle breakdown disruptions	A knowledge-based modelling approach involving Policies, Algorithm and Modeling.
15	Albareda-Sambola *et al.* (2014)	Dynamic Multi-period VRP	Probabilistic service requests	Adaptive service policy
16	Armas and Melian-Batista (2015)	DVRP with time windows	Customers requests	VNS
17	Schyns (2015)	DVRPTW, split delivery and heterogeneous fleet	Stochastic information	ACO
18	Ferrucci and Bock (2015)		New Customers requests	Penalty cost method
19	Mavrovouniotis and Yang (2015)	General DVRP	Travel time	Ant Colony Optimization
20	Euchi *et al.* (2015)	DVRP with pickup and delivery	Service requests	Artificial Ant Colony based on 2-Opt local search (AAC_2-Opt)

2.3 Common Solution Techniques for FLP and VRP

The aim of every optimization method is to obtain the best available solution from the feasible solution space and quite a number of methods have been proposed and described in literature for solving the FLP optimally. Many of these methods generate inexact solutions as experiments have shown that the FLP falls in the class of NP-hard combinatorial problems, that is, problems for which there are no known polynomial time algorithm for their solution. In fact, Krarup and Pruzan (1983) verified that the uncapacitated plant location problem, a simple FLP, is NP-hard. For this reason, exact methods are usually sacrificed for inexact methods to arrive at near-optimal solutions. Thus, there are two categories of techniques available for FLP: exact and heuristics (approximate) methods. In many cases, however, solutions that started with exact methods are usually concluded with heuristic approaches to obtain desirable optimal solutions.

Heuristics can be viewed as methods of arriving at approximate solution to a problem, that is, getting a good enough guessed solution to a problem. Thus, making use of heuristic methods in finding the solution of a problem is to apply a rule of thumb which is generally under the control of computer to explore available paths and reasonable guesses to arrive at an optimal solution. It is a search mechanism that checks all available alternatives with the goal of obtaining the best. Heuristics are problem-dependent. In other words, a heuristic is usually defined for the particular problem it seeks to solve. The meta-heuristics, a form of heuristics, differ from the basic heuristics in that they are problem-independent techniques and can generally be adapted to different types of problems. In the literature, a number of these approximate methods have been proposed to provide near optimal solutions to the FLPs. Below are some of the widely used methods for solving facility location related problems.

Branch-and-bound: The branch-and-bound (BB) is by far the most widely used exact method for solving large-scale NP-hard optimization problems (Clausen, 1999). The BB algorithm searches the space of the feasible solution of a given problem for the best solution. Since the number of possible solutions grow exponentially, only an implicit search of the solution space is possible. At the start of the process, only one subset of the solution space exists and is the complete solution space having an optimal solution set at ∞. In a successive manner, a pool of unexplored subsets are generated and represented as nodes in a search tree. These nodes are then processed by an iterative algorithm having the following components: node insertion, estimation of bounds and branching.

This method is popular with many authors and has been incorporated into many software for suitable problem instances. Kim and Kim (2010) applied the method to

determine the locations of long-term care facilities by using some dominance properties that can identify partial solutions dominated by other solutions and subsequently removing such dominated solutions from the solution space. Beresnev (2013) proposed a BB algorithm for finding an optimal non-cooperative solution to a competitive FLP where competing parties open their facilities successively with the aim of capturing a good number of customers thereby maximizing their profits.

Constructive and Local Search: Constructive search algorithms build an optimal solution to a problem from the scratch by repeatedly extending the current known solution until a complete solution is found. The method generally improves the solution than random methods. It is often used to initialize many meta-heuristics for obtaining near optimal solutions. The algorithm is usually thought of as being fast as they are often a single-pass approach (Burke & Kendall, 2005).

Local search methods on their part, consider the neighborhood of the existing current solution for possible replacement. That is, a local search algorithm takes a complete solution and tries to improve it through local moves within the neighborhood. Often times, constructive algorithms are used to initiate solutions which local search heuristics build on to provide optimal solutions. Generally, a local search follows the hill-climbing search process until the best solution is reached.

Tabu Search: The tabu search (TS) is a meta-heuristic approach that allows a form of hill-climbing to overcome local search optimal (Glover, 1989). The technique is based on the neighborhood search of the solution space in which the most recent moves are set in a tabu list, also known as the short term memory of the search, in a way that prevents movement in cycles (Abyasi-Sani & Ghanbari, 2016). In this way, moves in subsequent iterations which take the current solution to points in the feasible space which have previously been explored are avoided.

The search space and its neighborhood structure are two basic elements of TS algorithm. When different definitions of the search space for a given problem are considered, the neighborhood structures inevitably change at a certain degree (Gendreau & Potvin, 2005). A typical example is the capacitated FLP where the search space may be defined in respect to the location variable. In this case, the neighborhood structure will usually involve moves that change the status of one location and open facility from one location to another.

To describe the algorithm, the following notations are assumed: $f(x)$ is the objective function, x is the current solution, x^* is the current best known solution, f^* is the value of f at x^*, $N(x)$ is the neighborhood of x, $\overline{N}(x)$ is the non-tabu (i.e., admissible) subset of $N(x)$ and T is the tabu list. A general algorithm to implement the search is

given below.

Step 1: Choose an initial solution x_0 and set $x = x_0$, $x^* = x_0$ so that $f^* = f(x_0)$, and $T = \emptyset$.

Step 2: While termination criterion is not satisfied, select

$$s \in argmin\{f(x^{'})|x^{'} \in \overline{N}(x)\}$$

Step 3: If $f(x) < f^*$, set $f^* = f(x)$, $x^* = x$ and record tabu for the current move in T. Oldest entry is deleted if necessary.

Step 4: Repeat process until an optimal solution is obtained.

(Adaptive) Large Neighborhood Search Algorithm: The large neighborhood search (LNS) is a meta-heuristic and was first proposed by Shaw (1998). A comprehensive description of the algorithm was provided in Pisinger and Ropke (2010). Unlike most neighborhood search methods where the neighborhood containing the solution is usually explicitly defined, the LNS makes use of a destroy-repair approach to define an implicit neighborhood of solution. The destroy technique removes part of a current solution while the repair method re-inserts the removed candidate solution(s), thus rebuilding the solution structure. In the LNS, a solution neighborhood $J(x)$ is defined for a solution x. This neighborhood can then be explored by the destroy method followed by the repair method. The algorithm is designed in such a way that prevents a search of the entire neighborhood. Instead, only a random sample of it is explored at a time. The LNS heuristic is appended the prefix large because at the destruct phase, a large portion of the solution can be removed.

In describing the LNS algorithm, three basic variables are usually considered: x_α- best observed solution during search, x- current solution, x_p- a solution under probation of destruct or repair, and two functions $d(\cdot)$ and $r(\cdot)$ corresponding to destroy and repair method respectively. It is important to consider carefully the degree of freedom allowed at the destruction phase of the algorithm. The removal of very small or large part of the solution may result in poor solution. Hence, a mild degree is usually desirable. The methods of solution removal and re-insertion are best applied to problems that can be decomposed to the main and sub-problems. At the destruction phase, an infeasible solution is produced and then converted to a feasible solution by the repair method. Thus, it is right to say an LNS alternates between infeasible and feasible solutions. The steps required to implement the algorithm are:

Step 1: Set $x_\alpha = x$

Step 2: Apply the destroy technique followed by the repair method to find a new temporal solution x_p. That is, $x_p = r(d(x))$.

Step 3: Accept an improved solution as the new current solution so that $x = x_p$

Step 4: If $z(x_p) < z(x_\alpha)$ ($z(x)$ is the value of the objective function at x), then $x_\alpha = x_p$

Step 5: Check if stopping criterion is satisfied. If yes, stop. If not, repeat the process from step 2.

The Adaptive large neighborhood search (ALNS) is an extension of LNS heuristic. The first proposition of this extension was given by Ropke and Pisinger (2006). This method allows multiple application of the destroy-repair method within the same search. A weight value that controls the frequency of attempt by each method is assigned. This assignment, however, undergoes dynamical re-adjustment as the search progresses in order to allow problem adaptation. The use of multiple destroy-repair method indicates that the ALNS allows the creation of multiple neighborhoods. An applicable algorithm is as follows:

Step 1: Initialize $x_\alpha = x$ and define $\theta_- = (1, \ldots, 1)$ and $\theta_+ = (1, \ldots, 1)$ as destroy and repair methods weight storing variables such that $\theta_- \in \Re^{|\mu^-|}$, $\theta_+ \in \Re^{|\mu^+|}$ where μ^- and μ^+ are sets of destroy and repair methods respectively.

Step 2: Destroy and repair methods $d \in \mu^-$ and $r \in \mu^+$ are selected using θ_- and θ_+ such that $x_p = r(d(x))$

Step 3: Accept an improved solution as current solution so that $x = x_p$

Step 4: If $z(x_p) < z(x_\alpha)$, set $x_\alpha = x_p$

Step 5: Update θ_- and θ_+

Step 6: Check if the stopping criterion is satisfied or return to step 2.

Particle Swarm Optimization: The first work on PSO was published by Eberhart and Kennedy (1995). The particle swarm optimization (PSO) being a form of stochastic optimization does not require an operator to extract a new set of candidate solution. This is sharply in contrast to most evolutionary search (ES) methods. It is however similar to ES heuristics in that it also conduct a search of the population of potential solution. Furthermore, unlike the mutation stage of the ES, the PSO relies on information exchange between particles in the population. One can therefore say, a PSO utilizes a set of agents, called particles, to search a solution space for optimum value of a given problem. These particles move at a trajectory determined by a rule that merge the current velocity of a particle, its exploration histories and neighbors

(Kameyama, 2009). Parsopoulos and Vrahatic (2002) described the PSO in the following way: assume an n-dimensional search space and $X_i = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$- the vector space of the i^{th} particle, f- index of lowest function valued particle (i.e., best particle of the swarm), $b_i = \{b_{i1}, b_{i2}, \ldots, b_{in}\}$- best previous position of the i^{th} particle $V_i = \{v_{i1}, v_{i2}, \ldots, v_{in}\}$- velocity of the i^{th} particle.

The updating rule of particles movement through the search space is given by the following equations representing their coordinates and velocity respectively:

$$X_i^{j+1} = X_i^j + V_i^{j+1}$$

and

$$V_i^{j+1} = \beta \left(w V_i^j + k_1 \tau_{i1}^j \left(b_i^j - X_i^j \right) + k_2 \tau_{i2}^j \left(b_f^j - X_i^j \right) \right)$$

$i = 1, 2, \ldots, S$; S is the size of the population. β is the velocities-controlling parameter; w is the inertial weight; k_1 and k_2 are called the cognitive and social parameters respectively; $\tau_{i1}, \tau_{i1} \in [0, 1]$ are uniformly distributed random numbers.

The convergence behavior of a PSO is well determined by an appropriate choice of w. To facilitate a search of new solution space, experimental results have indicated that this can be done by initializing w to a large value, while for fine-tuning results, a small w can be used. Hence, a gradual reduction of the weight value can generate a good result.

Ant Colony Optimization and Variants: The Ant colony optimization (ACO) is a metaheuristic inspired by the way real ants find shortest paths from their nest to food sources (Dorigo & Stützle, 2004). The ants communicate through the release of a chemical substance known as the pheromone. This substance influences the behavioural pattern of other ants in the same environment. The ACO makes use of a constructive algorithm to create a solution by a sequence of probabilistic decisions where every move extends an incomplete solution by adding a new solution component until a complete solution is obtained. The sequence of moves can be viewed as a path through a corresponding decision graph (Merkle & Middendorf, 2005).

The implementation of an ACO starts by initializing the values of the pheromones by which a solution is constructed through a transition rule. Once a solution is obtained, the set of local pheromone is updated immediately. Local search is then employed to search for an improved solution and based on the best global solution obtained at different iterations, the global pheromone information is updated accordingly.

Simulate Annealing: Simulated annealing (SA) is a technique adapted from metal annealing process (Santosa & Kresna, 2015). It is a probabilistic based method for finding the global minimum of a cost function that possesses more than one local minimum (Bertimas & Tsitsiklis, 1993). The SA algorithm comprises of a non-homogeneous discrete time Markov chain at current state $x(t)$ such that if $x(t) = i$, a neighborhood $j(i)$ is chosen at random. Given a finite set S, the probability that any $j \in S(i)$ is selected is q_{ij}. Once such j is selected, the next state $x(t+1)$ is computed as follows:

If $J(j) \leq J(i)$, then $x(t+1) = j$. However, if $J(j) > J(i)$ then $x(t+1) = j$ with probability

$$P[x(t+1)] = exp\left[\frac{-J(j) - J(i)}{T(t)}\right]$$

Otherwise, $P[x(t+1)] = i$.

In essence,

$$[P[x(t+1) = j|x(t) = i] = q_{ij} exp\left[\frac{-1}{T(t)} max\{0, J(j) - J(i)\}\right] \;\; \forall \;\; j \neq i, \; j \in S(i)$$

and

$$[P[x(t+1) = j|x(t) = i] = 0 \;\; if \;\; j \neq i, \; j \notin S(i)$$

where J is a real-valued cost function defined on S, set $S(i) \subset S - \{i\}$ for each $i \in S$ is the neighborhood of i, T is a non-increasing function such that $T : \mathbf{Z}^+ \rightarrow (0, \infty)$, called the cooling schedule and $T(t)$ is the temperature at time t. Other assumptions of the algorithm are $x(0) \in S$ and $j \in S(i)$ if and only if $i \in S(j)$.

If a homogeneous Markov chain denoted by $x_T(t)$ is considered where $T(t)$ is held constant, then the invariant probability distribution with normalizing constant Y_T is given as

$$I_T(i) = \frac{1}{Y_T} exp\left[-\frac{J(i)}{T}\right] \;\; \forall i \in S$$

Genetic Algorithm: Genetic algorithm (GA) mimics the natural evolution process (Bhattachariya, 2013). The algorithm makes use of major properties of evolutionary system such as the population of chromosomes, selection of chromosome according to fitness, production of new offspring through crossover, and the random mutation of the new offspring (Mitchell, 1995). The GA algorithm searches through a space of chromosomes which often take the form of a bit string with two possible values of 0 and 1

called the alleles, by changing from one population to another. In the search process, a fitness score is assigned to each chromosome in the current population. The score assigned is dependent on the solvability index of the chromosomes to the particular problem under study.

Generally, GA works with three major operators. The selection operator chooses chromosomes for reproduction according to their fitness scores. In short, the selection operator preserves the best current solution while keeping the size of the population constant. It does this by eliminating from the population the bad solutions and reproducing copies of the good ones simultaneously. The crossover operator mimics the natural recombination process between organisms by exchanging two chromosomes to reproduce two new offsprings. The mutation operator maintains the diversity from one generation of chromosome population to another by the introduction of new features into the solution pool. A comprehensive details on this technique and other meta-heuristics highlighted above can be found in Sean (2013).

2.4 Applications of Lagrangian Relaxation and Sub-gradient Optimization Methods to FLP and VRP

Usually, the optimal solution to a problem lies between a given bound that may be found by means of well-constructed algorithms. Often, researchers are content on finding a value for the lower bound which by further experiments can turn out to be the desired optimal solution. Few well-known and general techniques are available for finding this lower bound. One such technique is the linear programming relaxation which takes the integer (or mixed inter) programming formulation of a problem and then relax the integrality constraints on the decision variable(s). The outcome of doing this is a linear problem that can be solved to optimality by standard algorithms like the simplex method or some heuristic techniques.

Another technique commonly used is the Lagrangian relaxation (LR) which, according to many studies, is both powerful and rich in analysis for solving NP-hard combinatorial optimization problems. The main idea behind the method is to identify one or more "complicating" constraints (Fisher, 2004) and then attach a multiplier vector also known as the penalty cost to this set of constraints before adding it to the objective function (Nezhad *et al.*, 2013). The resulting problem is called the Lagrangian problem and the objective function is known as the Lagrangian dual function. Maximizing the dual function gives the best lower bound value on the objective function of the original problem. Relaxing an integer programming problem in this sense generally produces a easy-to-solve problem (Beasley, 1993).

It is easy to deduce from the paragraph above that there are two main concerns in the

application of LR. The first is the strategic choice of the set of constraints to relax. Often, this can be done by isolating one or more interesting sub-problems and then relaxing the other constraints as described above. Another way is to dualize a set of linking constraints (that is, set of constraints where two constraints are represented) into the objective function (Guignard, 2003). This later approach is common when applying Lagrangian relaxation to facility location problems. (See for instance, Contreras *et al.* (2009)).

The second concern involves the tactical process of finding and updating the numerical values of the Lagrangian multipliers. To achieve this, the sub-gradient optimization method has appeared to be a very useful tool, which takes advantage of the problem structure and construct a numerical scheme similar to most gradient methods. Comparing among different methods, Beasley (1993) suggested that the method will nearly always out-performs other applicable techniques.

The use of LR and sub-gradient optimization methods are not new to problems formulated as FLP. In fact, due to the NP-hard nature of this class of problem, many heuristics proposed for finding optimal solutions are usually based on initial solutions provided by solving the Lagrangian problem of the original problem.

The LR technique constitutes a very powerful technique for solving large-scale optimization problems by exploiting some structural characteristics of these problems and obtaining the optimal solution bounds. These bounds in most cases provide the core of many numerical methods and a starting solution for some heuristics (Litvinchev *et al.*, 2010). The relaxation of a problem in the Lagrangian sense entails taking a complicating constraint(s) from the set of constraints, attaching a penalty cost function to it, and then adding same to the objective function. This penalty function is also known as the Lagrange multiplier. In this way, an easy to solve Lagrangian problem is produced whose optimal value is a lower bound (in case of minimization problem) to the optimal value of the original problem (Fisher, 2004). This approach has been used to solve a reasonable number of hard problems including the capacitated FLP (Chen & Ting, 2008).

The capacitated clustering problem (CCP) is a variant of FLP that has been widely studied due to its wide range of real life application. For instance Koskosidis and Powell (1992) introduced it to consolidate customer orders into a vehicle shipment. For this problem, Yang *et al.* (2011) used the approach of LR to dualize the assignment constraints giving rise to a problem that decomposed into a 0-1 independent knapsack problem. This problem was then solved by the subgradient optimization in two phases.

Tragantalerngsak *et al.* (2000) proposed a LR-based branch-and-bound algorithm to solve a two-level FLP. In their application of LR, two constraints representing the assignment restrictions were relaxed into the objective function resulting into two sub-problems in the

space of the decision variables. One of these sub-problems contain a single variable while the other has two variables. Nezhad *et al.* (2013) followed the same approach to obtain two sub-problems for the single-source multi-product variant of FLP. These problems were solved by a local search enriched Lagrangian heuristics. An earlier work similar to this where choice of facility is allowed was studied by Mazzola and Neebe (1999). The sub-problems in this case are the uncapacitated FLP and the 0-1 knapsack problem. The study presented a sub-gradient method based on the information from the Lagrangian problem. Chen and Ting (2008) described a method that combines a Lagrangian heuristic with ant colony system to solve a variant of FLP with capacitated single source. For this problem, each customer is expected to be served by a single facility. The LR process involves relaxing the demand constraints(Chen & Ting, 2008).

Due to the size of many VRP, authors are always compelled to use heuristic techniques to solve the problems. However, because of the interesting advantages of LR, a few authors still approached the problem using this method. One of the most successful application of Lagrangian relaxation was conducted by Kohl and Madsen (1997). The Lagrangian dual problem of these authors was solved by combining the sub-gradient optimization and bundle methods. The sub-gradient method is, like most gradient methods, an iterative procedure for updating the Lagrangian multipliers until a tolerance criterion is satisfied. In Kallenhauge *et al.* (2006), the set of constraints that ensures each customer is served by exactly one vehicle was relaxed to yield a constrained shortest path sub-problem which was then solved by a cutting plane algorithm.

2.5 Identification of Gaps

In this chapter, comprehensive surveys have been reported on FLP and VRP, with the applications of LR and Sub-gradient methods for finding the optimal solutions of these classes of problem. Observation from the literature reviewed showed the following which motivated the research conducted in this thesis.

[i] The work by Ghiani *et al.* (2012) proposed a collection system that encourages on-site separation and collection of different wastes. However, their proposed system does not consider the quantity of wastes of each type from individual cluster. This particular gap is addressed in the model presented in this work.

[ii] Oduro-Kwarteng (2011) carried out an empirical study of the factors that influence the utilization and productivity of waste collection vehicles. Among the factors identified are the quality of roads, traffic and vehicle conditions. The study revealed that more respondents rated the condition of roads used for waste collection as bad. It was indicated that some

access routes were bad and that breakdowns of collection vehicles were frequent for vehicles that were not designed for such routes. It further claimed that bad roads had adverse effect on the performance of the waste collection vehicles. Road conditions limit the maximum quantity of waste to be carried by each vehicle type. However, the study did not suggest any mathematical approach to ensure waste vehicles utilize only good roads. Therefore, drawing conclusion from this study and the review conducted and reported in the earlier sections and to the best of my knowledge, no author has designed a disposal system that includes the attributes of roads. This new variable has been added by introducing a parameter called accessibility ratio.

[iii] The use of LR and sub-gradient methods in literature for solving these problems is substantial and many authors have not seized the many advantageous structures of the Lagrangian dual function. These methods were used analytically in this thesis.

CHAPTER THREE

METHODOLOGY

3.1 Lagrangian Relaxation

Lagrangian relaxation (LR) is a powerful tool for solving many hard combinatorial problems. LR provides bounds on the optimal solution of an IP, and the solution which is usually not feasible for the primal (original) problem, can often serve as a starting point for some specialized Lagrangian heuristics. Before proceeding into the full description of the LR, a definition of relaxation is first given. According to Geoffrion (1974), the relaxation of a minimization problem (rP_{min}) is given as follows.

Definition 3.1.1. *Problem* $(rP_{min}) : min[g(x)|x \in S]$ *is a relaxation of problem* $(P_{min}) : min[f(x)|x \in T]$*, possessing the same decision variable* x*, if and only if*

1. *The feasible set of* (rP_{min}) *is a superset of* (P_{min})*, i.e.,* $S \supseteq T$ *and*
2. *The objective function of* (rP_{min})*, over the feasible set of* (P_{min})*, dominates that of* (P_{min})*, i.e.,* $\forall x \in T$*,* $g(x) \leq f(x)$*.*

It is clear from this definition that $v(rP_{min}) \leq v(P_{min})$, where $v(P)$ is the optimal value of a problem (P).

In a similar version, if the original problem is a maximization problem, $(P_{max}) : max[f(x)|x \in T]$, a relaxation of (P_{max}) is a problem (rP_{max}) over the same decision variable x of the form $(rP_{max}) : max[g(x)|x \in S]$ such that

1. The feasible set of (rP_{max}) contains that of (P_{max}), i.e., $S \supseteq T$ and
2. The objective function of (rP_{max}), over the feasible set of (P_{max}), dominates that of (P_{max}). That is, for all $x \in T$, $g(x) \geq f(x)$ and consequently, $v(rP_{max}) \geq v(P_{max})$.

Relaxing a problem in the manner defined above plays largely two functions. They provide bounds on the optimal value of complex problems, while relaxed solutions, which are usually not feasible for the original problem, can often be used as starting points for building heuristics that provides information about the structure of the optimal solution (Beasley, 1993).

Since the optimal solution to a problem usually lies within a given bound (the lower and upper bounds), it is always very important to construct algorithms that address the quality of these bounds. Few well-known and general techniques are available for finding lower bounds. One such technique is the linear programming relaxation which takes an integer (or

mixed integer) programming formulation of a problem and then relax the integrality function of the variable. The result is a linear problem solvable by standard algorithms or heuristic methods. The other and commonly used in literature is the Lagrangian relaxation. In its entirety, it involves attaching a multiplier vector called the Lagrangian multipliers to some of the constraint of an integer (or mixed integer) programming formulation of a problem and then solving the resulting problem to optimality. Of course, as mentioned earlier, the solution gives a lower bound on the optimal solution of the original problem. Beasley (1993) pointed out that by absorbing some set of constraints in the objective function, many complex, NP-hard combinatorial problems can be reduced to simpler problems. LR, according to him as by practical experience, gives very good lower bounds at reasonable computational cost.

The choice of values for the multipliers determines the quality of the lower bound (the closer the lower bound, the better) and therefore choosing an appropriate method of making this choice is very vital to the success of the overall usage of the LR method. Two general approaches of doing this are the Lagrangian dual ascent also known as the multiplier adjustment technique and the sub-gradient optimization (SO). The SO is used in this work as it provides a stronger lower bound. For now we dwell more on LR, more details on SO will be provided in subsequent section.

Consider an IP problem defined as

$$\textbf{IP} \qquad z = min\{fx | Ax \leq b,\ Cx \leq d,\ x \in X,\ x_i \in I(X)\}$$

A and C are matrices of conformable dimensions, b, d, and f are vectors, and $I(X)$ is an index set denoting integer variables. The restrictions on the signs of x are contained in set X, and due to the nature of the problem, the integrality condition could be any of the following: $X = R_+^{n-m} \times R^m$, $X = R_+^{n-m} \times R_+^m$ or $X = R_+^{n-m} \times \{0, 1\}$, where $n \geq m \geq 1$. The set is assumed to be the set of indices $I(X)$ restricted to integer (or binary).

Definition 3.1.2. *Suppose $Ax \leq b$ is a complicating constraint, i.e., its absence would render the problem much easier to solve. If constraints $Cx \leq d$ and $x \in X$ are kept so that the problem (IP) is relaxed in the Lagrangian sense relative to the complicating constraint, with non-negative Lagrangian multipliers λ, then the Lagrangian relaxation formulation of (IP) is given as*

$$\boldsymbol{LR} \qquad z(\lambda) = min\{fx + \lambda(Ax - b) | Cx \leq d,\ x \in X\}$$

In the above formulation, the constraint $Ax \leq b$ has been dualized, i.e., its slacks have been added up to the objective function with weights λ, a process which consequently dropped it from the constraint list. It therefore follows from Definition 3.1 that

(i) The feasible set of LR contains that of IP

(ii) $fx + \lambda(Ax - b)$ for any value of x is feasible for (IP) and $\lambda \geq 0$

From (i) and (ii) above, it implies that $v(LR) \leq v(IP)\ \forall \lambda \geq 0$. To obtain the best lower bound on $v(IP)$, one solves the Lagrangian dual of (IP) given by

$$\textbf{LD} \qquad w_D = \max_{\lambda \geq 0}[v(LIP)] = \max_{\lambda \geq 0} \begin{pmatrix} min & [fx + \lambda(Ax - b)] \\ s.t & Cx \leq d \\ & x \in X \end{pmatrix}$$

Whereas LR is in the space of x, LD is in the dual space of λ.

Recall that the constraint equation of the form $Ax = b$ can be represented by a pair of inequality $Ax \leq b$ and $-Ax \leq -b$. To dualize such into the objective function, two appropriate multipliers μ and ν are chosen such that the problem

$$(IP)_1 \qquad min\,[fx]\,|Ax = b,\ \ Cx \leq d\ \ x \in X$$

now becomes

$$(LIP)_1 \qquad min\,[fx + \mu(Ax - b) + \nu(-Ax + b)]\,|Cx \leq d\ \ x \in X$$

where $\mu \geq 0$ and $\nu \geq 0$. $(LIP)_1$ can also be written as

$$(LIP)_1 \qquad min\,[fx + \lambda(Ax - b)]\,|Cx \leq d\ \ x \in X$$

where $\lambda = \mu - \nu$

The Lagrangian dual in this case is given as

$$(LIP)_{1D} \qquad \max_{\lambda \geq 0}[u(LIP)] = \max_{\lambda \geq 0} \begin{pmatrix} min & [fx + \lambda(Ax - b)] \\ s.t. & Cx \leq d \\ & x \in X \end{pmatrix}$$

where $u(\cdot)$ represents the optimal solution.

If $x(\lambda)$ is a Lagrangian solution of IP for some $\lambda \geq 0$, it is not always guaranteed that such solution (which is feasible for the dual problem) is also optimal for the original problem. Usually, since $fx(\lambda)$ is the feasible solution of IP, the optimal value of IP, $v(IP)$, satisfies the inequality $f(x)(\lambda) + \lambda[Ax(\lambda) - b] \leq v(IP) \leq fx(\lambda)$. Should $\lambda[Ax(\lambda) - b] = 0$, then $fx(\lambda) = v(IP) = fx(\lambda)$. This is called the complementary slackness condition.

The above assertions constitute a set of sufficient (but not necessary) conditions for the existence of an optimal solution, the summary of which is given below as contained in

Guignard (2003).

1. If $x(\lambda)$ is an optimal solution of LR for some $\lambda \geq 0$, then, $fx(\lambda)+\lambda[Ax(\lambda)-b] \leq v(IP)$
2. Suppose further that $x(\lambda)$ is feasible for IP, then $fx(\lambda)+\lambda[Ax(\lambda)-b] \leq v(IP) \leq fx(\lambda)$
3. Additionally, if $\lambda[Ax(\lambda) - b] = 0$, then $x(\lambda)$ is an optimal solution of IP, and $v(IP) = fx(\lambda)$.

The following results in Wolsey (1998) establish the relationship between IP and LR.

Proposition 3.1.1. *LR is a a relaxation of IP for all* $\lambda \geq 0$.

Proof. Recall that LR is a relaxation of IP if (i) The feasible solution region of LR is at least as large as that of IP. This is true since $\{x|Ax \leq b,\ x \in X\}$. (ii) The objective value is at least as great in LR as in IP for all feasible solutions in IP. As $\lambda \geq 0$ and $Ax \leq b$ for all $x \in X$, $fx + \lambda(Ax - b) \geq fx$ for all $x \in X$. □

Proposition 3.1.2. *If* $\lambda \geq 0$ *(i)* $x(\lambda)$ *is an optimal solution of LR (ii)* $Ax(\lambda) \leq b$*, and (iii)* $(Ax(\lambda))_i = b_i$ *whenever* $\lambda_i > 0$*. Then* $x(\lambda)$ *is optimal in IP.*

Proof. By (i) $w_D \leq z(\lambda) = fx(\lambda)+\lambda(Ax(\lambda)-b)$. By (iii) $fx(\lambda)+\lambda(Ax(\lambda)-b) = fx(\lambda)$. By (iii), $x(\lambda)$ is feasible in IP and so $fx(\lambda) \leq z$. This, $w_D \leq fx(\lambda)+\lambda(Ax(\lambda)-b) = fx(\lambda) \leq z$. But as $w_D \geq z$,equality holds throughout, and $x(\lambda)$ is optimal in IP. □

Next, some relevant properties of Lagrangian dual (LD) functions are stated with proofs where necessary. These are the structures that make LR applicable to IP problems. The first result which is from Geoffrion (1974) gives the geometric interpretation of LD in the primal space of $x \in X$.

3.1.1 Some Properties of Lagrangian Dual Function

PROPERTY I: Geometric Interpretation of LD

Theorem 3.1.3. *LD is equivalent to the primal relaxation given by*

$$\boldsymbol{PR} \qquad \min_x\{fx|Ax \leq b,\ x \in Co\{x \in X|Cx \leq d\}\}$$

in the sense that

$$v(LD) = v(PR)$$

where $Co(X)$ *is the convex hull of the set* X *(See section 1.8 for definition).*

PROPERTY II: The weak and strong Lagrangian dualities: The two problems IP and LD are said to form a weak dual pair if $z(\lambda) \leq fx$ for all $x \in X$ and $\lambda \geq 0$. When $z(x) = z(\lambda)$, they form a strong dual pair. Weak duals can be easily constructed from IP problems by taking the LP relaxation of the IP.

Theorem 3.1.4 (Weak Lagrangian Duality)**.** *Let IP, LR and LD be as defined above and x be a feasible solution of IP. Then, for $\lambda \geq 0$,*

$$z(\lambda) \leq fx \qquad \textit{and} \qquad z(\lambda) \leq z^* \leq \theta^*$$

where θ^* is the solution to LP relaxation of IP.

The theorem implies that the feasible solution of LD and its maximum value is a lower bound of the optimal value of IP. The proof can be found in Bertsekas *et al.* (2003).

Theorem 3.1.5 (Strong Lagrangian Duality)**.** *Suppose IP, LR and LD retain their respective definitions, if x^* solves the sub-problem LR for some $\lambda^* \geq 0$ and in addition, $Ax^* \geq b$ and $\lambda(b - Ax^*) = 0$, then x^* is an optimal solution of IP and λ^* is an optimal value of the LD.*

This theorem implies that if x^* solves LR and the dualized constraints are satisfied, then x^* and λ^* are the optimal values of IP and LD, respectively.

Generally, however, there is no guarantee that x^* and λ^* may be obtained such that $fx^* = z(\lambda^*)$. Hence, there is always a relative percentage gap between the best solution of IP and LD. When this occurs, then it is said that a duality gap exists in the solution to the problem. The duality difference given by $\Delta^* = x^* - \lambda^*$ is the absolute value of the duality gap. If the duality gap exists and the constraint set X is discrete, then the dual function is typically non-differentiable. This special feature makes the SO method suitable for solving the LD of IP problems.

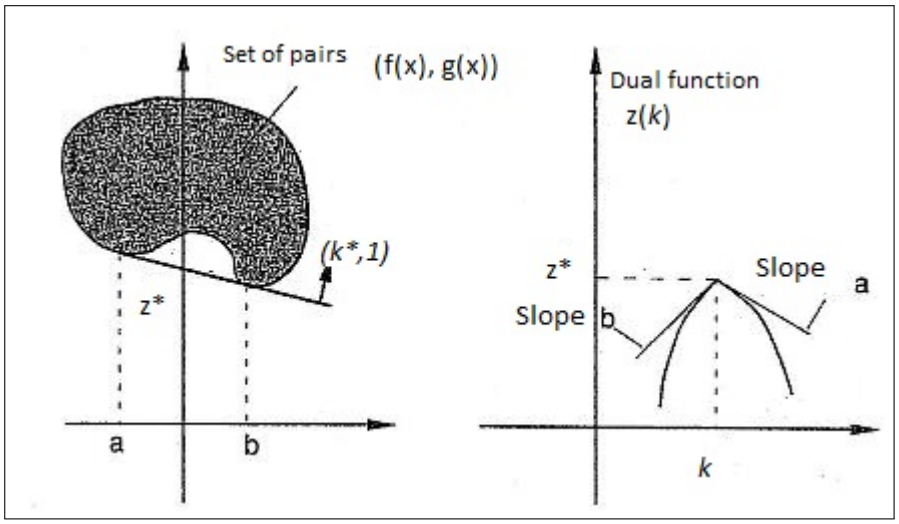

Figure 3.1: An illustration of the non-differentiabilty of LD when there is duality gap. $g(x)$ is the discrete set of constraints.

Source: Reprinted from Convex Analysis and Optimization (page 21), by D. P. Bertsekas *et al.*, 2003, USA: Athena Scientific. Copyright [2002] by Dimitri P. Bertsekas. Reprinted with permission.

The next property shows how strong the bound of a LD is. In fact, the lower bound provided by the LD is at least as large as the lower bound obtained from the relaxation of IP (Guignard, 2003).

PROPERTY III:

Theorem 3.1.6 (The strength of the bound of a LD problem.)**.** *Suppose IP, LR and LD maintain the definitions above and let z^* be the optimal value of LD, then*

$$z^* = min\ \ fx\ \ s.t\ \ Ax \leq b,\ \ x \in Co(X)$$

(Geoffrion (1974); Guta (2003)). $X \subseteq Z_+^n$ and bounded $\Rightarrow X$ consists of a finite number of points, say $X = \{x_1, x_2, \ldots, x_p\}$

$$\begin{aligned} z^* &= \max_{\lambda \geq 0}\{z(\lambda)\} \\ &= \max_{\lambda \geq 0}\{min\ fx + \lambda(Ax - b) | x \in X\} \\ &= \max_{\lambda \geq 0}\{min\ fx_i + \lambda(Ax_i - b) | i = 1, \ldots, p\} \\ &= max\{\mu | \mu \leq fx_i + \lambda(Ax_i - b) | i = 1, \ldots, p; \mu \in \mathbf{R}^1, \lambda \in \mathbf{R}_+^n\} \end{aligned} \tag{3.1}$$

where μ is the lower bound of $\{fx_i + \lambda(Ax_i - b) | i = 1, \ldots, p\}$. Problem (3.1) is a LP problem with variables $(\mu, \lambda) \in \mathbf{R}^1 \times \mathbf{R}_+^n$. The dual of (3.1) gives

$$z^* = min \sum_{p=1}^{P} \alpha_p (fx_p)$$

$$suchthat$$

$$\sum_{p=1}^{P} \alpha_p (Ax_p - b) \geq 0$$

$$\sum_{p=1}^{P} \alpha_p = 1$$

$$\alpha_p \geq 0;\ \ p = 1, \ldots, P$$

Setting $x = \sum_{p=1}^{P} \alpha_p x_p$, $\sum_{p=1}^{P} \alpha_p = 1$ with $\alpha_p \geq 0$ for each $p = 1, \ldots, P$, the result is

$$z^* = minfx$$

$$suchthat$$

$$Ax \geq b,\ \ x \in Co(X)$$

as required. □

It is obvious from the proof that the theorem is based on the definition of the convex hull of the set X.

PROPERTY IV: Integrality Property of Lagrangian Relaxation

The LR is said to possess the integrality property if $Co\{x \in X | Cx \leq d\} = \{x | Cx \leq d\}$. That is, the extreme points of $\{x | Cx \leq d\}$ are in X. In short, when the integrality condition holds, $z^*_{LD} = z^*_{LP}$ where z^*_{LD} is the optimal value of LD and z^*_{LP} is the optimal value of LP.

3.1.2 Construction of Lagrangian Relaxation

The right application of LR to IP problems depend largely on two quality decisions: (i) the choice of the set(s) of constraints to relax and (ii) the tactical process of finding the numerical values of λ. In making a choice of which constraint to relax, usually one may choose among the following options: isolating an interesting sub-problem and relaxing the other constraints; dualizing linking constraints (i.e., those constraints where decision variables are connected); if more than one interesting sub-problem exists with common variables, one may split these variables and then dualize a copy of the constraint; and sometimes aggregate copies of the constraint may be dualized rather than individual copy. The following examples illustrate the first two techniques.

Example 3.1.1: Consider the generalized assignment problem formulated as an IP problem as

$$\phi = min \left[\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} x_{ij} \right] \tag{3.2}$$

such that

$$\sum_{i=1}^{m} x_{ij} = 1, \qquad j = 1, \ldots, n \tag{3.3}$$

$$\sum_{j=1}^{n} a_{ij} x_{ij} \leq b_i, \qquad i = 1, \ldots, m \tag{3.4}$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \ldots, m; \;\; j = 1, \ldots, n \tag{3.5}$$

Two natural LR exists for this problem. The first is obtained by dualizing constraints (3.3) with $\lambda \geq 0$ as the Lagrangian multiplier, so that

$$\begin{aligned}
\phi(\lambda) &= min\left[\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}x_{ij} + \sum_{j=1}^{n}\lambda_j\left(\sum_{i=1}^{m} x_{ij} - 1\right)\right] \\
&= min\left[\sum_{i=1}^{m}\sum_{j=1}^{n}[x_{ij}(f_{ij}+\lambda_j) - \lambda_j]\right] \\
&\quad suchthat \\
&\quad \sum_{j=1}^{n} a_{ij}x_{ij} \leq b_i, \qquad i = 1,\ldots,m \\
&\quad x_{ij} \in \{0,1\} \quad i = 1,\ldots,m;\ \ j = 1,\ldots,n
\end{aligned}$$

Hence, the problem reduces to a m 0-1 knapsack problem which can be solved in time proportional to $n\sum_{i=1}^{m} b_i$

The second relaxation is obtained by dualizing (3.4), this time with $\mu \geq 0$ so that,

$$\begin{aligned}
\phi(\mu) &= min\left[\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}x_{ij} + \sum_{i=1}^{m}\mu_i\left(\sum_{j=1}^{n} a_{ij}x_{ij} - b_i\right)\right] \\
&= min\left[\sum_{j=1}^{n}\left(\sum_{i=1}^{m}(f_{ij}+\mu_i a_{ij})x_{ij}\right) - \sum_{i}^{m}\mu_i b_i\right] \\
&\quad suchthat \\
&\quad \sum_{i=1}^{m} x_{ij} = 1, \qquad j = 1,\ldots,n \\
&\quad x_{ij} \in \{0,1\} \quad i = 1,\ldots,m;\ \ j = 1,\ldots,n
\end{aligned}$$

Constraint set (3.3) is an upper bound constraint and thus the LR problem can be viewed as an upper bound problem solvable in time proportional to mn by evaluating $min[f_{ij} + \mu_i a_{ij}]$ for each j and setting the associated $x_{ij} = 1$ in conformity with (3.3). The remaining x_{ij} are set to zero according to (3.5).

This example above illustrates how LR may be constructed by taking advantage of interesting simpler sub-problems in a given "hard" problem.

The next example illustrates the construction of LR by dualizing constraints that link two or more decision variables.

Example 3.1.2: (Nezhad *et al.*, 2013) Consider the mathematical formulation of the uncapacitated single source multi-product FLP

$$\varphi = min\left[\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p}\gamma_{ijk}x_{ijk} + \sum_{i=1}^{m}\sum_{k=1}^{p} f_{ik}y_{ik}\right] \tag{3.6}$$

such that

$$\sum_{i=1}^{m} x_{ijk} = 1, \;\; j = 1, \ldots, n; k = 1, \ldots, p. \tag{3.7}$$

$$x_{ijk} \leq y_{ik}, \;\; i = 1, \ldots, m; j = 1, \ldots, n; k = 1, \ldots, p. \tag{3.8}$$

$$\sum_{k=1}^{p} y_{ik} \leq 1, \;\; i = 1, \ldots, m. \tag{3.9}$$

$$x_{ijk}, \; y_{ik} \in \{0, 1\}, \;\; i = 1, \ldots, m; j = 1, \ldots, n; k = 1, \ldots, p. \tag{3.10}$$

The two decision variables are x_{ijk} and y_{ik}. The two variables are linked in constraints (3.8). If this set of constraints is relaxed into the objective function (3.6), two sub-problems are obtained, each in the space of the individual variable. The LD for this problem is as follows:

$$\varphi(\lambda) = min \left[\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} (\gamma_{ijk} + \lambda_{ijk}) \, x_{ijk} + \sum_{i=1}^{m} \sum_{k=1}^{p} \left(f_{ik} - \sum_{j=i}^{n} \lambda_{ijk} \right) y_{ik} \right]$$

$$\begin{aligned} &suchthat \\ &\sum_{i=1}^{m} x_{ijk} = 1, \;\; j = 1, \ldots, n; k = 1, \ldots, p. \\ &\sum_{k=1}^{p} y_{ik} \leq 1, \;\; i = 1, \ldots, m. \\ &x_{ijk}, \; y_{ik} \in \{0, 1\}, \;\; i = 1, \ldots, m; j = 1, \ldots, n; k = 1, \ldots, p. \end{aligned}$$

The first sub-problem in the space of x variable gives

$$\varphi_x(\lambda) = min \left[\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} (\gamma_{ijk} + \lambda_{ijk}) \, x_{ijk} \right]$$

$$\begin{aligned} &suchthat \\ &\sum_{i=1}^{m} x_{ijk} = 1, \;\; j = 1, \ldots, n; k = 1, \ldots, p. \\ &x_{ijk} \in \{0, 1\}, \;\; i = 1, \ldots, m; j = 1, \ldots, n; k = 1, \ldots, p. \end{aligned}$$

Similar to Example 3.1.1, this sub-problem can be solved by evaluating $min(\gamma_{ijk} + \lambda_{ijk})$ for each $j = 1, \ldots, n$ and $k = 1, \ldots, p$, and setting $x_{ijk} = 1$. The second sub-problem in the

space of y reduces to an assignment problem of the form

$$\varphi_y(\lambda) = min\left[\sum_{i=1}^{m}\sum_{k=1}^{p}\left(f_{ik} - \sum_{j=i}^{n}\lambda_{ijk}\right)y_{ik}\right]$$

$$suchthat$$

$$\sum_{k=1}^{p} y_{ik} \leq 1, \;\; i = 1, \ldots, m; k = 1, \ldots, p.$$

$$y_{ik} \in \{0, 1\}, \;\; i = 1, \ldots, m; j = 1, \ldots, n; k = 1, \ldots, p.$$

This approach of relaxation is particularly suitable to problems complicated with constraints with linked variables. As will be shown in the succeeding chapter, the mathematical models proposed in this thesis are characterized by this kind of constraints, hence, the approach illustrated in this example was adopted to obtain a number of sub-problems.

The other concern, as mentioned earlier, in the application of LR to IP problems is the evaluation of the Lagrangian multiplier. Quite a few techniques have been developed for this purpose among which are: column generation, dual ascent, cutting plane (constraint generation), bundle method, etc. (see Guignard (2003) for details). However, the sub-gradient remain the most widely used in literature. This is because most dual functions are non-differentiable (concave) functions, a feature that the method utilizes to generate iteratively the values of the multipliers.

3.2 Sub-gradient Optimization

Before going into the details of the sub-gradient optimization method (SOM), we first show that the dual function $z(\lambda)$ is concave.

Theorem 3.2.1. *The dual function $z : \boldsymbol{R}^n \rightarrow \boldsymbol{R}$, defined by*

$$z(\lambda) = min\{fx + \lambda(Ax - b) | x \in X\}$$

is concave.

Proof. Let $\lambda_1, \lambda_2 \in \mathbf{R}^n$ and $\alpha \in [0, 1]$. Then

$$z(\lambda_1) = min\{fx + \lambda_1(Ax - b) | x \in X\}$$

and

$$z(\lambda_2) = min\{fx + \lambda_2(Ax - b) | x \in X\}$$

For $\overline{\lambda} = \alpha\lambda_1 + (1-\alpha)\lambda_2$,

$$\begin{aligned} z(\overline{\lambda}) &= min\{fx + \overline{\lambda}(Ax - b) | x \in X\} \\ &= f\overline{x} + \overline{\lambda}(A\overline{x} - b), \text{ for some } \overline{x} \in X \\ &= \alpha[f\overline{x} + \lambda_1(A\overline{x} - b)] + (1-\alpha)[f\overline{x} + \lambda_2(A\overline{x} - b)] \\ &\geq \alpha z(\overline{\lambda}) + (1-\alpha) z(\overline{\lambda}) \end{aligned}$$

which verifies that $z(\lambda)$ is concave. □

A sub-gradient algorithm that solves a dual concave function like $z(\lambda)$ above is an iterative method which, from an initial set of multipliers, generates in a systematic fashion optimal Lagrangian multipliers (Beasley, 1993). Given the value of the current multiplier vector, say λ^k at iteration k, the SOM process involves taking a step along a sub-gradient of $z(\lambda^k)$. As earlier stated, the method works for non-differentiable dual functions. Hence, unlike the gradient methods, in the SOM, the gradients are replaced by the sub-gradients. Next, the formal definition of sub-gradient is given.

Definition 3.2.1. *Let $g : \Re^n \to \Re$ be concave. The vector $s \in \Re^n$ is a sub-gradient of g at $y \in \Re^n$ if*

$$g(y) + s(x - y) \geq g(x) \ \forall\, x \in \Re^n$$

The set of all sub-gradients of g at y is called the sub-differential of $g(y)$ and is given by

$$\partial g(y) = \{s | g(y) + s(x - y) \geq g(x) \ \forall\, x \in \Re^n$$

If $\partial g(y) \neq \emptyset$, then g is said to be differentiable at y.

The following theorem establishes the condition for a point $x^* \in \Re^n$ to be the value of LD defined for the function g.

Theorem 3.2.2. *A necessary and sufficient condition for $x^* \in \Re^n$ to be the optimal value of a concave function g over $\Re^n$ is $0 \in \partial g(x^*)$.*

Proof. By Definition (3.3), $0 \in \partial g(x^*)$ for $x^* \in \Re^n$ if and only if

$$g(x) - g(x^*) \leq 0(x - x^*) \ \forall\, x \in \Re^n \iff g(x) \leq g(x^*) \ \forall\, x \in \Re^n$$

□

The iterative scheme for the SOM is given by

$$\lambda^{k+1} = E_\Omega(\lambda^k + \mu_k s^k), \;\; k = 0, 1, \ldots \tag{3.11}$$

where s^k is a sub-gradient of $z(\lambda^k)$, $\mu_k \geq 0$ is the step size and $E_\Omega(\cdot)$ is the Euclidean projection on the feasible set Ω.

The following following result from Polyak (1969) sets the approximate limit on μ_k.

Theorem 3.2.3. *Let λ^* be an optimal solution to $\max_{x\in\Omega}\{z(x)\}$ where z is a concave function over $\Re^n$ and Ω is a closed convex subset of $\Re^n$. If*

$$0 < \mu_k < \frac{2\left(z(\lambda^*) - z(\lambda^k)\right)}{\|s^k\|^2} \tag{3.12}$$

then,

$$\|\lambda^{k+1} - \lambda^k\| < \|\lambda^k - \lambda^*\|$$

where $\|\cdot\|$ denotes the Euclidean norm.

Proof.

$$\begin{aligned}
\|\lambda^{k+1} - \lambda^*\|^2 &= \|E_\Omega(\lambda^k + \mu_k s^k) - \lambda^*\|^2 \\
&\leq \|\lambda^k + \mu_k s^k - \lambda^*\|^2 \\
&= \|\lambda^k - \lambda^*\|^2 + 2\mu_k s^k(\lambda^k - \lambda^*) + \mu_k^2\|s^k\|^2 \\
&= \|\lambda^k - \lambda^*\|^2 + \mu_k\left[\mu_k\|s^k\|^2 - 2\left[s^k(\lambda^* - \lambda^k)\right]\right] \\
&\leq \|\lambda^k - \lambda^*\|^2 + \mu_k\left[\mu_k\|s^k\|^2 - 2\left[z(\lambda^*) - z(\lambda^k)\right]\right] \\
&< \|\lambda^k - \lambda^*\|^2
\end{aligned}$$

Since $s^k(\lambda^* - \lambda^k) \geq z(\lambda^*) - z(\lambda^k)$ (due to the concavity of z) and $\mu_k\|s^k\|^2 - 2\left[z(\lambda^*) - z(\lambda^k)\right] < 0$ by (3.12).

Therefore,

$$\|\lambda^{k+1} - \lambda^k\| < \|\lambda^k - \lambda^*\|$$

□

In practice, the value of μ_k is usually estimated by

$$\mu_k = \frac{t_k \left(z_{ub} - z(\lambda^k)\right)}{\|s^k\|^2} \tag{3.13}$$

where t_k is the step size parameter, such that $0 < t_k \leq 2$ and z_{ub} is the upper bound on the dual function $z(\lambda)$. The justification for (3.13) is given in Held *et al.* (1974). Initial guessed values of $t_0 = 2$ and $\lambda^0 = 0$ have worked very well for a number of problems (Fisher (2004), Contreras *et al.*, (2009)) by halving the value of t_k whenever $z(\lambda)$ has failed to increase in some fixed number of iterations. The step size parameter t_k controls the size of the steps along the direction of s^k.

Suppose all the definitions above hold with the constraints $Ax_i \leq b$, $x_i \in I(X)$ still relaxed, the basic steps of a SOM algorithm are:

Step 1: Let t_0 be such that $0 < t_0 \leq 2$. Initialize Z_{ub}. Choose an initial value of λ^k.

Step 2: Solve the LD with the current set λ^k to get a solution X_k of value z_{lb} (lower bound value of Z).

Step 3: Define sub-gradients s^k for the relaxed constraints evaluated at the current solution by

$$s^k = Ax_k - b \ \forall \ i = 1, \ldots, m$$

Step 4: Define a scalar step size μ_k by (3.13). μ_k depends on the difference $z_{ub} - z(\lambda^k)$ and the value of t_k. $\|s^k\|^2$ is the scaling factor.

Step 5: Update λ^k using $\lambda^k = max\left(0, \lambda^k + \mu_k s^k\right)$ and return to Step 2 to re-solve LD with the new set of λ^k.

The convergence of this algorithm requires that μ_k satisfies the following two conditions (Guta, 2003):

$$\lim_{k \to \infty} \mu_k = 0 \qquad and \qquad \sum_{k=0}^{\infty} \mu_k = \infty \tag{3.14}$$

Theorem 3.2.4 (Convergence of the sub-gradient algorithm)**.** *Consider the LD problem* $max\{z(\lambda) | \lambda \in \Omega = \Re_+^n$ *where* z *is the dual function and bounded from above on* Ω *so that*

$$\Omega^* = \{\overline{\lambda} \in \Omega | z(\overline{\lambda}) \geq z(\lambda), \ and \ \lambda \in \Omega\} \neq \emptyset$$

If $\{\lambda^k\} \subseteq \Omega$ *is a sequence of points generated by the recursive formula (3.11) where* $s^k \in$

$\partial z(\lambda^*)$ is given by $s^k = Ax^k - b$ for some $x^k \in X^*(\lambda^k)$ and $\lambda^k \geq 0$ satisfies (3.14), then $z(\lambda^k) \to z^* = z(\lambda^*)$ where $\lambda^* \in \Omega^*$.

Proof. **Proof:** To prove this result, it will be shown that for any arbitrary $\epsilon > 0$, $\exists\ J_0 > 0$ such that $j \geq J_0 \Rightarrow z(\lambda^*) - z(\lambda^k) < \epsilon$. Suppose by contradiction that $\exists\ \epsilon > 0$ such that

$$z(\lambda^*) - z(\lambda^k) \geq \epsilon\ \ \forall\ k \tag{3.15}$$

Suppose $x^k \in X^*(\lambda^k)$ so that $s^k = Ax^k - b \in \partial z(\lambda^k)$ for each k. By definition

$$z(\lambda^k) + s^k(\lambda - \lambda^*) \geq z(\lambda)\ \ \forall\ k$$

setting $\lambda = \lambda^*$ and using (3.15)

$$s^k(\lambda^* - \lambda^k) \geq \epsilon$$

Multiplication with a negative factor $-2\mu_k$ gives

$$2\mu_k s^k(\lambda^k - \lambda^*) \leq -2\mu_k\epsilon$$

Thus,

$$\begin{aligned}
\|\lambda^{k+1} - \lambda^*\|^2 &= \|E_\Omega(\lambda^k + \mu_k s^k) - \lambda^*\|^2 \\
&\leq \|\lambda^k + \mu_k s^k - \lambda^*\|^2 \\
&= \|\lambda^k - \lambda^*\|^2 + \mu_k^2\|s^k\|^2 + 2\mu_k s^k(\lambda^k - \lambda^*) \\
&\leq \|\lambda^k - \lambda^*\|^2 + \mu_k^2\|s^k\|^2 - 2\mu_k\epsilon
\end{aligned}$$

Since X is finite such that $X = \{x^h | h = 1, 2, \ldots, H\}$, set

$$\|s^*\|^2 = max\{\|s^h\|^2 | s^h = Ax^h - b,\ \ h = 1, 2, \ldots, H\}$$

By the first part of (3.14), a value $J_1 > 0$ can be found so that

$$\mu_k \leq \frac{\epsilon}{\|s^*\|^2}\ \ \forall\ k \geq J_1$$

That is,

$$\mu_k\|s^*\|^2 \leq \epsilon$$

Hence,

$$\begin{aligned}\|\lambda^{k+1} - \lambda^*\|^2 &= \|\lambda^k - \lambda^*\|^2 + \mu_k \left(\mu_k \|s^k\|^2 - 2\epsilon\right) \\ &\leq \|\lambda^k - \lambda^*\|^2 - \mu_k \epsilon, \ \ \forall\, k \geq J_1\end{aligned}$$

For any arbitrary $M \geq J_1$, the last inequality gives

$$0 \leq \|\lambda^{M+1} - \lambda^*\|^2 \leq \|\lambda^{J_1} - \lambda^*\|^2 - \epsilon \sum_{k=J_1}^{M} \mu_k \tag{3.16}$$

According to (3.14), $\sum_{k=J_1}^{M} \mu_k \to \infty$ as $M \to \infty$, therefore,

$$\|\lambda^{J_1} - \lambda^*\|^2 - \epsilon \sum_{k=J_1}^{M} \mu_k \to -\infty$$

which is a contradiction. □

3.3 Model Formulation

In this section, the two models proposed in this thesis are presented. The formulation of these models followed the research tools described in chapter one, that is, the IP problems.

3.3.1 Formulation of Solid Waste Collection Model

The main objective of this model was to find the optimal location of waste collection facilities in a residential area. The residential area is partitioned into zones, called the clusters, each of which is assigned to a collection facility drawn from a set of candidate facilities. These assignments are based on certain restrictions on the capacity of waste that a facility can accommodate and the distances between the clusters and facilities. Another feature of the model is the allocation of waste containers to open facilities. Collectively, three decisions are made: what number of collection facilities should be activated; which cluster should be assigned to a candidate facility and how many containers of various types should be allocated to an activated facility. The various assumptions in the model are given below, some of which were based on the work of Ghiani *et al.* (2012).

1. All variables are integers. In particular, the location-allocation variables are binary integers.

2. All sets are finite.

3. The total capacity of containers at an open facility must exceed the quantity of waste shipped to the facility.
4. All clusters must be assigned to the activated facilities such that the capacity of each facility is not exceeded.
5. At least one container of each waste type is allocated to an activated facility.
6. The distance between a cluster and a facility must be within a threshold distance.
7. All distances are measured in meter.

Model sets, variables and parameters

The mathematical model presented in this work was based on the following sets, decision variables and input parameters, with their definition given appropriately.

Sets

C is the set of customers i, $i \in C$

S is the set of candidate waste collection site k, $k \in S$ and $S \subset C$

T is the set of containers of different types t, $t \in T$.

Decision Variables

$$x_k = \begin{cases} 1 \text{ if a facility is set up at a candidate site } k \in S; \\ 0 \text{ otherwise.} \end{cases}$$

$$y_{ik} = \begin{cases} 1 \text{ if a customer } i \in S \text{ is assigned to a candidate site } k \in S; \\ 0 \text{ otherwise.} \end{cases}$$

z_{tk} = a positive integer representing the number of containers of type $t \in T$ assigned to collection site $k \in S$

Input Parameters

n_t is the total number of containers of type $t \in T$ available for allocation.

β_{ti} is the shipment size of waste type $t \in T$ by cluster of candidate hub $i \in C$

c_t is the capacity of each container of type $t \in T$

d_{ik} is the distance between a cluster of candidate hub $i \in C$ and a collection site $k \in S$

D is the threshold distance defined as the maximum allowable distance between a cluster and a candidate site.

μ is an arbitrary integer value constant which is greater than or equal to the total number of clusters.

The mathematical formulation of the problem is as follows:

$$Min \quad \sum_{k \in S} x_k$$

is the objective function and it minimizes the total number of activated collection sites with the following constraints

$$\sum_{k \in S} y_{ik} = 1 \qquad \forall \quad i \in C \tag{3.17}$$

ensures that each cluster $i \in C$ is assigned to one and only one activated site $k \in S$;

$$\sum_{k \in S} z_{tk} \leq n_t \qquad \forall \quad t \in T \tag{3.18}$$

prevents the total number of containers of type $t \in T$ assigned to site $k \in S$ to exceed the total available containers for waste of type $t \in T$;

$$\sum_{i \in C} \sum_{t \in T} \beta_{ti} y_{ik} \leq \sum_{t \in T} c_t z_{tk} \qquad \forall \quad k \in S \tag{3.19}$$

ensures that the total quantity of waste of type $t \in T$ from a cluster $i \in C$ directed to site $k \in S$ does not exceed the total capacity of containers of type $t \in T$ allocated to $k \in S$;

$$\mu x_k \geq \sum_{t \in T} z_{tk} \qquad \forall \quad k \in S, \quad \mu \geq |T| \tag{3.20}$$

enforces assignment of containers only to sites that are open. There is no need to ensure that at least one container is located at each activated site, because the number of sites is minimized. Hence, if a site does not have any containers assigned, it will not be open;

$$d_{ik} y_{ik} \leq D \qquad \forall \quad i \in C, \quad k \in S \tag{3.21}$$

ensures that allocating a cluster $i \in C$ to a site $k \in S$ is only possible when the distance between any such cluster and site does not exceed the value of D;

$$x_k \in \{0, 1\} \qquad \forall k \in S \tag{3.22}$$

$$y_{ik} \in \{0, 1\} \qquad \forall i \in C, \quad \forall k \in S \tag{3.23}$$

$$z_{tk} \geq 0; \qquad \forall t \in T, \quad k \in S \tag{3.24}$$

are the integrality constraints.

3.3.2 Formulation of Solid Waste Disposal Model

In this section, the mathematical formulation of the solid waste collection vehicle routing problem with time windows, kickbacks and road attributes (SVRPTWKR) is examined. In the problem, a cost optimal set of routes was obtained for a homogeneous fleet of vehicles such that the requests at the activated collection sites are satisfied within their specified time windows. In addition, the problem takes into consideration the kickbacks during everyday operation. Kickbacks in this context include the lunch and rest breaks together. There are three possible periods within which a kickback may occur: (i) immediately after serving the request at a node, (ii) between traveling from a node to another node, (iii) immediately before serving the request of a node. The most important feature of this model was the inclusion of constraints on road attributes. It becomes necessary to add this limitation due to the deplorable state of roads in most developing countries which has increased the number of damaged vehicles used for waste collection (see Oduro-Kwarteng (2011)).

This problem is described on a graph $G = (N, A)$. N and A are the vertices and edges of routes respectively where $N = D_0 \cup F \cup S$ is the set of all nodes in the system, where D_0 is the depot, $F = \{1, \ldots, m\}$ is the set of m waste recycling facilities (WRF), and $S = \{m+1, \ldots, m+n\}$ is the set of n waste collection sites (WCS), $k \in S$. $A = \{(f, k) : f, k \in N, f \neq k\}$ is the set of arcs linking all the nodes in the system. Other parameters of the model are defined as follows:

$V = \{1, \ldots, v\}$ is the set of homogeneous vehicles, $h \in V$

T is the set of different waste types, $t \in T$

ψ_t is the capacity of each compartment of the vehicle

s_k is the service time at node $k \in S$

$[\sigma_k, \lambda_k]$ is the time window in which the request of $k \in S$ can be satisfied

q_{tk} is the quantity of waste of type $t \in T$ picked up at node $k \in S$

a_{kh} is the start time of service for vehicle $h \in V$ at node $k \in S$

r_{kth} is the cumulative request of waste of type $t \in T$ at node $k \in S$ for vehicle $h \in V$

τ_{fk} is the travel time along arc (f, k)

w_{fk} is the distance associated with arc (f, k)

p_d is the kick back duration that must occur within the time window $[\sigma^d, \lambda^d]$

ρ_h is the ratio of driving between previous and next stop when vehicle $h \in V$ observes a break

R_{fk} is the relative accessibility ratio of arc (f,k)

M is any large constant or the longest distance between any two nodes

$\phi_{fkh} = 1$ if vehicle $h \in V$ uses arc (f,k); 0, otherwise

$\varphi_{fkh} = 1$ when vehicle $h \in V$ observes a break along arc (f,k) and 0, otherwise

$\theta_{fk} = 1$ if $R_{fk} > 0.5$ for the arc (f,k); 0, otherwise

The objective of the model is given as

$$Min \left[\sum_{(f,k)\in A} w_{fk}\theta_{fk} \left(\sum_{h\in V} \phi_{fkh} \right) \right]$$

subject to the following constraints

$$\sum_{k\in F\cup S} \phi_{0kh} = 1 \qquad \forall \quad h \in V \tag{3.25}$$

$$\sum_{f\in F\cup S} \phi_{f0h} = 1 \qquad \forall \quad h \in V \tag{3.26}$$

$$\sum_{f\in N}\sum_{k\in N} \phi_{fkh} = 1 \qquad \forall \quad h \in V \tag{3.27}$$

$$\sum_{f\in N} \phi_{fkh} - \sum_{k\in N} \phi_{kfh} = 0 \quad \forall \quad k \in F \cup S \quad h \in V \tag{3.28}$$

$$\sigma_f \le a_{fh} \le \lambda_f \qquad \forall \quad f \in N, \quad h \in V \tag{3.29}$$

$$\sum_{t\in T} r_{0th} = 0 \qquad \forall \quad h \in V \tag{3.30}$$

$$r_{fth} + q_{ft} \le r_{kth} + (1 - \phi_{fkh})M \qquad \forall \quad f \in N - F, \quad k \in N, \quad h \in V \tag{3.31}$$

$$r_{fth} \le c_t \qquad \forall \quad f \in N, \quad t \in T, \quad h \in V \tag{3.32}$$

$$r_{fth} \ge 0 \qquad \forall \quad f \in N, \quad t \in T, \quad h \in V \tag{3.33}$$

$$\sum_{(f,k)\in A} \varphi_{fkh} = 1 \qquad \forall \quad h \in V \tag{3.34}$$

$$\varphi_{fkh} \le \phi_{fkh} \qquad \forall \quad (f,k) \in A \quad h \in V \tag{3.35}$$

$$a_{fh} + s_k + \varphi_{fkh}p^d + \tau_{fk} \le a_{kh} + (1 - \phi_{fkh})M \qquad \forall \quad (f,k) \in A \quad h \in V \tag{3.36}$$

$$\sigma^d + p^d + \tau_{fk}(1 - \rho_h) \le a_{kh} + (1 - \varphi_{fkh})M \qquad \forall \quad (f,k) \in A \quad h \in V \tag{3.37}$$

$$a_{fh} + s_k + \tau_{fk}\rho_h \le \lambda^d + (1 - \varphi_{fkh})M \qquad \forall \quad (f,k) \in A \quad h \in V \tag{3.38}$$

$$0 \le \rho_h \le 1 \qquad h \in V \tag{3.39}$$

$$\theta_{fk} \leq |A| \qquad \forall \quad (f,k) \in A \tag{3.40}$$

$$\sum_{(f,k)\in A} \theta_{fk}\phi_{fkh} \geq 1 \qquad h \in V \tag{3.41}$$

$$\phi_{fkh} \in \{0,1\} \qquad \forall \quad (f,k) \in A \quad h \in V \tag{3.42}$$

$$\varphi_{fkh} \in \{0,1\} \qquad \forall \quad (f,k) \in A \quad h \in V \tag{3.43}$$

$$\theta_{fk} \in \{0,1\} \qquad \forall \quad (f,k) \in A \tag{3.44}$$

Constraints (3.25) and (3.26) ensure that all vehicles start and end at the depot. In constraints (3.27), each node is served exactly once while (3.28) ensure equal inflow and outflow except at the depot. Inequalities (3.29) and (3.36) describe the time windows and service time constraints. Equations (3.30) ensure vehicles are empty at the beginning of operation. Accumulation of demand for all nodes except at the depot is handled by constraints (3.31). The capacity of each vehicle is prevented from being exceeded by constraint (3.32). (3.33) are non-negativity constraints. By constraints (3.34), each vehicle must observe exactly one kickback while in (3.35) kickbacks only occur between connected nodes. Constraints (3.37) and (3.38) are the time windows for kickbacks. Constraints (3.39) defines the interval of ρ_h. By constraints (3.40), number of executable routes cannot exceed the number of possible route in the network while (3.41) ensures that only routes with high attributes are considered. Constraints (3.42), (3.43) and (3.44) are the binary decision variables.

Summarily, we have proposed two models that are based on simple clustering and the introduction of a two new parameters on road attributes. While the first handled the phase of waste collection, the second deals with the disposal of waste from collection sites to final disposal centers. The method of solution to be adopted is based on the Lagrangian relaxation technique. Results in in the previous sections were used to establish that the resulting dual functions are concave. To obtain the bounds on the optimal values of the decision variable, the LD function is solved within the SOM algorithm.

3.4 Analysis of Solid Waste Collection Model

The SWC model, as earlier mentioned, captures a simple clustering approach to locating a set of potential waste collection sites in a residential area such that the distance between each cluster and each potential site does not exceed a pre-defined threshold distance. In the IP formulation of the problem, there were three decision variables denoted by x_k, y_{ik} and z_{tk}. These variables were linked by the constraint sets (3.19) and (3.20). To obtain the LR to this problem, these constraints are relaxed into the objective function in the following

manner:

$$
\begin{aligned}
L(v,w) &= Min\left[\sum_{k\in S} x_k + \sum_{k\in S} v_k\left(\sum_{i\in C}\sum_{t\in T}\beta_{ti}y_{ik} - \sum_{t\in T} c_t z_{tk}\right) + \sum_{k\in S} w_k\left(\sum_{t\in T} z_{tk} - \mu x_k\right)\right] \\
&= Min\left[\sum_{k\in S} x_k - \sum_{k\in S}\mu w_k x_k + \sum_{t\in T} z_{tk}\left(\sum_{k\in S} w_k - c_t\sum_{k\in S} v_k\right) + \sum_{k\in S} v_k\sum_{i\in C}\sum_{t\in T}\beta_{ti}y_{ik}\right] \\
&= Min\left[\sum_{k\in S} x_k\left(1-\mu w_k\right) + \sum_{k\in S} v_k\sum_{i\in C}\sum_{t\in T}\beta_{ti}y_{ik} + \sum_{t\in T} z_{tk}\left(\sum_{k\in S} w_k - c_t\sum_{k\in S} v_k\right)\right]
\end{aligned}
$$

subject to

$$
\begin{aligned}
&\sum_{k\in S} y_{ik} = 1 \qquad \forall \quad i\in C \\
&\sum_{k\in S} z_{tk} \le n_t \qquad \forall \quad t\in T \\
&d_{ik}y_{ik} \le D \qquad \forall \quad i\in C, \quad k\in S \\
&x_k \in \{0,1\} \qquad \forall \quad k\in S \\
&y_{ik} \in \{0,1\} \qquad \forall \quad i\in C, \quad \forall k\in S \\
&z_{tk} \ge 0; \qquad \forall \quad t\in T, \quad k\in S
\end{aligned}
\tag{LR}
$$

LR above can be separated into three sub-problems each in the space of the decision variables. The sub-problem in the space of the x-variable corresponds to

$$L_x(v,w) = Min\left[\sum_{k\in S} x_k\left(1-w_k\right)\right]$$

subject to

$$x_k \in \{0,1\} \qquad \forall \quad k\in S$$

The solution to L_x constitutes a set of optimal activated waste collection sites. The problem can be solved by a simple inspection of the sign of $(1-w_k)$. If $(1-w_k) \le 0$, a solution $x^* = 1$ is obtained; otherwise, $x^* = 0$

The sub-problem in the space of y-variable corresponds to

$$L_y(v,w) = Min\left[\sum_{k\in S} v_k\sum_{i\in C}\sum_{t\in T}\beta_{ti}y_{ik}\right]$$

subject to

$$\sum_{k\in S} y_{ik} = 1 \qquad \forall \quad i\in C$$

$$d_{ik} y_{ik} \leq D \qquad \forall \quad i \in C, \quad k \in S$$

$$y_{ik} \in \{0,1\} \qquad \forall \quad i \in C, \quad \forall k \in S$$

L_y is an assignment problem which can further be decomposed into two simpler sub-problems if the assignment constraint is relaxed (see Section 3.1.2). The solution to this sub-problem gives the bounds on the optimal assignment of clusters to the potential sites.

The sub-problem in the space of z is given by

$$L_z(v,w) = Min \left[\sum_{t \in T} z_{tk} \left(\sum_{k \in S} w_k - c_t \sum_{k \in S} v_k \right) \right]$$

Subject to

$$\sum_{k \in S} z_{tk} \leq n_t \qquad \forall \quad t \in T$$

$$z_{tk} \geq 0; \qquad \forall \quad t \in T, \quad k \in S$$

L_z is a typical unbounded knapsack problem (Pisinger (1995)) and its optimal solution gives the number of each type of container to be allocated to each open site.

Using Theorem 3.5, it can be shown that the dual functions resulting from the three sub-problems above are concave, the proof of which will validate the use of the SOM.

Theorem 3.4.1. *The dual function* $L_x : \Re^n \rightarrow \Re$ *defined as*

$$L_x(v,w) = Min \left[\sum_{k \in S} x_k \left(1 - \mu w_k\right) \right]$$

is concave.

Proof. Let $w^1, w^2 \in \Re^n$ and $\alpha \in [0,1]$. Then

$$L_x(w^1) = Min \left[\sum_{k \in S} x_k \left(1 - \mu w_k^1\right) \right]$$

and

$$L_x(w^2) = Min \left[\sum_{k \in S} x_k \left(1 - \mu w_k^2\right) \right]$$

Suppose $\overline{w} = \alpha w^1 + (1-\alpha)w^2$, then

$$\begin{aligned} L_x(\overline{w}) &= Min\left[\sum_{k\in S} x_k\left(1-\mu\overline{w}_k\right)\right] \\ &= \sum_{k\in S}\overline{x}_k\left(1-\mu\overline{w}_k\right) \quad \text{for some } \overline{x}_k \geq 0 \\ &= \alpha\left[\sum_{k\in S}\overline{x}_k\left(1-\mu w_k^1\right)\right] + (1-\alpha)\left[\sum_{k\in S}\overline{x}_k\left(1-\mu w_k^2\right)\right] \\ &\geq \alpha L_x(\overline{w}) + (1-\alpha)L_x(\overline{w}) \end{aligned}$$

□

In a similar way it can be shown that $L_y(v,w)$ is concave. Next, the concavity of $L_z(v,w)$ is established.

Theorem 3.4.2. *The dual function $L_z : \Re^n \to \Re$ defined by*

$$L_z(v,w) = Min\left[\sum_{t\in T} z_{tk}\left(\sum_{k\in S} w_k - c_t\sum_{k\in S} v_k\right)\right]$$

is concave.

Proof. **Proof:** Let $v^1, v^2 \in \Re^n$, $w^1, w^2 \in \Re^n$ and $\alpha \in [0,1]$. Then

$$L_z(v^1,w^1) = Min\left[\sum_{t\in T} z_{tk}\left(\sum_{k\in S} w_k^1 - c_t\sum_{k\in S} v_k^1\right)\right]$$

and

$$L_z(v^2,w^2) = Min\left[\sum_{t\in T} z_{tk}\left(\sum_{k\in S} w_k^2 - c_t\sum_{k\in S} v_k^2\right)\right]$$

Now for $\overline{v} = \alpha v^1 + (1-\alpha)v^2$ and $\overline{w} = \alpha w^1 + (1-\alpha)w^2$ we obtain

$$\begin{aligned} L_z(\overline{v},\overline{w}) &= Min\left[\sum_{t\in T} z_{tk}\left(\sum_{k\in S}\overline{w}_k - c_t\sum_{k\in S}\overline{v}_k\right)\right] \\ &= \sum_{t\in T}\overline{z}_{tk}\left(\sum_{k\in S} w_k - c_t\sum_{k\in S} v_k\right) \quad \text{for some } \overline{z}_{tk} \geq 0 \\ &= \alpha\left[\sum_{t\in T}\overline{z}_{tk}\left(\sum_{k\in S} w_k^1 - c_t\sum_{k\in S} v_k^1\right)\right] + (1-\alpha)\left[\sum_{t\in T}\overline{z}_{tk}\left(\sum_{k\in S} w_k^2 - c_t\sum_{k\in S} v_k^2\right)\right] \\ &\geq \alpha L_z(\overline{v},\overline{w}) + (1-\alpha)L_z(\overline{v},\overline{w}) \end{aligned}$$

Hence, the dual function $L_z(v, w)$ is concave. □

The next result shows that the sum of these dual functions is also concave.

Theorem 3.4.3. *Let L_x, L_y and L_z defined above be real-valued concave functions over the domain $\Re^n$. A function L, so that for all $\phi = (v, w) \in \Re^n$, defined as $L(v, w) = L_x(v, w) + L_y(v, w) + L_z(v, w)$, is concave.*

Proof. Since $L_x(v, w)$, $L_y(v, w)$ and $L_z(v, w)$ are concave with common domain $\Re^n$, then for all $\phi, \psi \in \Re^n$ and $\alpha \in [0, 1]$, we have the following inequalities

$$L_x[\alpha\phi + (1-\alpha)\psi] \geq \alpha L_x(\phi) + (1-\alpha)L_x(\psi)$$

$$L_y[\alpha\phi + (1-\alpha)\psi] \geq \alpha L_y(\phi) + (1-\alpha)L_y(\psi)$$

and

$$L_z[\alpha\phi + (1-\alpha)\psi] \geq \alpha L_z(\phi) + (1-\alpha)L_z(\psi)$$

From these inequalities and the definition of of L, the following is obtained

$$\begin{aligned} L[\alpha\phi + (1-\alpha)\psi] &= L_x[\alpha\phi + (1-\alpha)\psi] + L_y[\alpha\phi + (1-\alpha)\psi] + L_z[\alpha\phi + (1-\alpha)\psi] \\ &\geq \alpha L_x(\phi) + (1-\alpha)L_x(\psi) + \alpha L_y(\phi) + (1-\alpha)L_y(\psi) \\ &+ \alpha L_z(\phi) + (1-\alpha)L_z(\psi) \\ &= \alpha[L_x(\phi) + L_y(\phi) + L_z(\phi)] + (1-\alpha)[(1-\alpha)L_x(\psi) \\ &+ (1-\alpha)L_y(\psi) + (1-\alpha)L_z(\psi)] \\ &\geq \alpha L(\phi) + (1-\alpha)L(\psi) \end{aligned}$$

□

which shows that L is a concave function which is solvable using the SOM method described in chapter three. A sub-gradient s corresponding to the relaxed constraints (3.19) and (3.20) with respect to the Lagrangian multiplier vector (v, w) is given by

$$s^k = \left[\left(\sum_{i \in C} \sum_{t \in T} \beta_{ti} y_{ik} - \sum_{t \in T} c_t z_{tk} \right)_k \right]$$

The SOM solves the LD problem given by

$$D = \max_{v,w \geq 0} L(v, w)$$

A scheme to implement the SOM for problem D is as follows:

Algorithm 3.1: A Sub-gradient Optimization Scheme for SWC Model

Step 1: Initialize the SO parameters as follow: $v = 0$, $w = 0$ (Lagrangian multipliers), $m = 1$ (iteration counter), $t_k : 0 < t_k \leq 2$ (the user-defined control parameter), $l = 0$ (lower bound value), $u = N$ (upper bound value).
Repeat process until $m = m_{max}$ (maximum number of iterations).

Step 2: Solve the sub-problems $L_x(v, w)$, $L_y(v, w)$ and $L_z(v, w)$ with $v_k = v_k^{m-1}$, $w_k = w_k^{m-1}$.

Step 3: Compute a new lower bound as

$$L(v_k^{m-1}, w_k^{m-1}) = L_x(v_k^{m-1}, w_k^{m-1}) + L_y(v_k^{m-1}, w_k^{m-1}) + L_z(v_k^{m-1}, w_k^{m-1})$$

Step 4: **If** the Lagrangian solution $(x^{*m-1}, y^{*m-1}, z^{*m-1})$ is feasible for (IP), i.e.,

$$\sum_{i \in C} \sum_{t \in T} \beta_{ti} y_{ik}^{*m-1} \leq \sum_{t \in T} c_k z_{tk}^{*m-1} \quad \text{and} \quad \mu x_k^{*m-1} \geq \sum_{t \in T} z_{tk}^{*m-1}$$

Step 5: Update the lower and upper bound values as $l = u = l(v^{m-1}, w^{m-1})$ and stop. The Lagrangian solution is an optimal solution for IP.
Else begin

Step 6: **If** $l(v^{m-1}, w^{m-1}) > l$, **then** update $l : l = l(v^{m-1})$

Step 7: **Else if** $u(v^{m-1}, w^{m-1}) < u$, **then** update $u : u = u(v^{m-1}, w^{m-1})$.

Step 8: Update the Lagrangian multipliers:

$$v_k^m = v_k^{m-1} + \sigma_m \left[\sum_{i \in C} \sum_{t \in T} \beta_{ti} y_{ik}^{*m-1} - \sum_{t \in T} c_k z_{tk}^{*m-1} \right]$$

and

$$w_k^m = w_k^{m-1} + \sigma_m \left[\sum_{t \in T} z_{tk}^{*m-1} - \mu x_k^{*m-1} \right]$$

where σ is the scalar step size given by

$$\sigma = \frac{t(u - l)}{\sum_{k \in S} (s^k)^2}$$

End Else.

Step 9: Increment the counter $m = m + 1$.

This algorithm was implemented on the AMPL (A Modeling Language for Mathematical Programming) and solved with the embedded CPLEX (originally for **simplex method** as implemented on **C Programming Language**) solver. The direct computation was done on the AIMMS (Advanced Interactive Multidimensional Modeling System).

3.5 Relaxation, Reformulation and Analysis of SWD Model

The model proposed for the transportation phase of SWCD has two major contributions. One is the addition of a variable and parameter that decide the accessibility of a road to a vehicle. The other contribution is that at the various activated collection sites, as observed from model I, different containers are hosted for collecting different types of waste. This translates to the on-site-sorting procedure and thus requires either a multi-compartment vehicle or specially designated vehicles to pick up these wastes and haul them to the nearest disposal or treatment site. This problem, however, furnished with these two extra features, becomes harder to solve considering the NP-hard nature of the classical VRP itself and the limited computing resources available. Hence, a relaxed version of the problem is solved.

To obtain the relaxed problem, the kickback variables and the associated parameters and constraints were dropped. Furthermore, the disposal of a single waste type is considered at a time in a single-vehicle-single-depot system. This problem may be formulated as follows. Consider a directed graph $G = (N, A)$ where $N = \{0\} \cup S$ is the set of nodes to be visited. $\{0\}$ denotes the depot where a fleet of collection vehicles is hosted and S is the set of collection sites to be visited. $A = \{(i, j)|i, j \in N\}$ is the set of arcs that a vehicle can use for its daily operations. Each collection site has an associated request q_i (quantity of waste ready for collection) and a time window $[e_i, l_i]$ representing the interval of time within which the request of site $i \in N$ can be served: e_i and l_i are the lower and upper bound values of this time windows. The distance between each node is given by d_{ij} and the service time at $i \in N$ is denoted by s_i. For each arc (i, j), there is an associated travel time t_{ij}. Obviously, initial route elimination can be enabled by setting $l_i + t_{ij} > l_j$ and $q_i + q_j > C$, where C is the vehicle capacity.

There are four main decision variables: $\phi_{ij}|(i, j) \in A$ equal 1 if the vehicle uses arc (i, j) and 0 otherwise; $\theta_{ij}|(i, j) \in A$ equal 1 if $R_{ij} > b$ and 0, otherwise. Recall that R_{ij} is the accessibility ratio of arc $(i, j) \in A$ and b is a scalar constant such that $0 \leq b \leq 1$. The other variables are a_i, the arrival time of vehicle at node $i \in N$ and r_i is the load of the vehicle arriving at node $i \in N$. M is a large constant (that may be taken as the largest distance between any two nodes). The MIP formulation of the model is given as follows.

$$\textbf{CASE I} \qquad \tau = \text{Min}\left[\sum_{(i,j)\in A} d_{ij}\phi_{ij}\theta_{ij}\right] \tag{3.45}$$

subject to

$$\sum_{j\in N}\phi_{ij} = 1 \qquad \forall \quad i \in N \tag{3.46}$$

$$\sum_{j\in N}\phi_{ij} - \sum_{j\in N}\phi_{ij} = 0 \qquad \forall \quad i \in N \tag{3.47}$$

$$a_i + s_i + t_{ij} \leq a_j + (1 - \phi_{ij}\theta_{ij})\, M \qquad \forall \quad (i,j) \in A \tag{3.48}$$

$$r_i + q_i \leq r_j + (1 - \phi_{ij}\theta_{ij})\, M \qquad \forall \quad (i,j) \in A \tag{3.49}$$

$$e_i \leq a_i \leq l_i \qquad \forall \quad i \in N \tag{3.50}$$

$$0 \leq r_i \leq C \qquad \forall \quad i \in N \tag{3.51}$$

$$\theta_{0j} = \theta_{i0} = 1 \qquad \forall \quad i,j \in N \tag{3.52}$$

$$\sum_{j\in N}\theta_{ij} \leq |A| \forall \quad i \in N \tag{3.53}$$

$$\sum_{j\in N}\phi_{ij}\theta_{ij} \geq 1 \qquad \forall \quad i \in N \tag{3.54}$$

$$\phi_{ij},\ \ \theta_{ij} = \{0,1\} \qquad \forall \quad (i,j) \in A \tag{3.55}$$

The objective function (3.45) minimizes the total distance covered by the vehicle. Constraints (3.46) prevent a vehicle from visiting a customer more than once while constraints (3.47) ensure that the vehicle starts and ends each trip at the depot. Constraints (3.48-3.50) define the time windows, whereas constraints (3.51) ensure the feasibility of the load at customer $i \in N$. Equations (3.52) impose that the arcs linking the depot to a customer is feasible and vice versa. Constraints (3.53) ensure that the number of executable arcs does not exceed the total number of arcs in the system. Constraints (3.54) ensure that only arcs with high attributes are used for the route construction. The integrality restrictions are imposed by (3.55).

The above formulation results in a nonlinear objective function in ϕ_{ij} and θ_{ij}. One may consider another case where θ_{ij} is dropped from the objective function. In this case, the

objective function is linear and is written as

$$\textbf{CASE II} \qquad \tau = \text{Min}\left[\sum_{(i,j)\in A} d_{ij}\phi_{ij}\right] \tag{3.56}$$

with constraints (3.46)-(3.55) still valid. However, to tighten MIPs, the nonlinear term may be given a new representation as follows: Let

$$\psi_{ij} = \phi_{ij}\theta_{ij} \tag{3.57}$$

Note that $\psi_{ij} = 1$ only if both ϕ_{ij} and θ_{ij} equal 1, and it equal 0 otherwise. Hence, by introducing a binary variable ψ_{ij} in the formulation, $\phi_{ij}\theta_{ij}$ is replaced accordingly by the addition of the following constraints:

$$-\psi_{ij} + \phi_{ij} + \theta_{ij} \leq 1 \qquad \forall \quad (i,j) \in A \tag{3.58}$$

$$2\psi_{ij} - \phi_{ij} - \theta_{ij} \leq 0 \qquad \forall \quad (i,j) \in A \tag{3.59}$$

$$\psi_{ij},\ \ \phi_{ij},\ \ \theta_{ij} = \{0,1\} \qquad \forall \quad (i,j) \in A \tag{3.60}$$

By constraints (3.58), it is required that $\psi_{ij} + 1 \geq \phi_{ij} + \theta_{ij}$. This forces ψ_{ij} to be equal to 1 whenever ϕ_{ij}, $\theta_{ij} = 1$. Constraints (3.59) ensure that $2\psi_{ij} \leq \phi_{ij} + \theta_{ij}$, thus permitting $\psi_{ij} \geq 0$ whenever both ϕ_{ij} and θ_{ij} equal 1. ψ_{ij} only assume the value of 0 when either ϕ_{ij} or θ_{ij} is 0.

As earlier stated, the VRP is NP-hard and its solution by direct exact method may sometimes fail and always time consuming especially for large-scale problems. Thus, by implication, the problem under consideration is NP-hard and in order to find its solution, the Lagrangian relaxation is first applied to reduce its complexity. The resulting problem is then solved by applying sub-gradient optimization.

Next, we consider the Lagrangian relaxation of constraint (3.46) which requires that the vehicle should visit each customers once. By assumption, (3.46) forms the set of complicating constraints. By dualizing them into the objective functions (3.45) and (3.56) with the Lagrangian multiplier $u_i \geq 0 \in \Re^n$, and assuming (3.57) holds, then the following Lagrangian problem is obtained.

$$LR_I \qquad \tau(u) = \text{Min}\left[\sum_{(i,j)\in A} d_{ij}\phi_{ij}\theta_{ij} + \sum_{i\in N} u_i\left(\sum_{j\in N}\phi_{ij} - 1\right)\right]$$
$$= \text{Min}\left[\sum_{i\in N}\sum_{j\in N}\left(d_{ij}\theta_{ij} + u_j\right)\phi_{ij} - \sum_{j\in N} u_i\right]$$

subject to (3.47-3.55) and (3.57-3.59)

Similarly,

$$LR_{II} \qquad \rho(u) = \text{Min}\left[\sum_{(i,j)\in A} d_{ij}\phi_{ij} + \sum_{i\in N} u_i\left(\sum_{j\in N}\phi_{ij} - 1\right)\right]$$
$$= \text{Min}\left[\sum_{i\in N}\sum_{j\in N}\left(d_{ij} + u_j\right)\phi_{ij} - \sum_{j\in N} u_i\right]$$

subject to (3.47-3.55) and (3.57-3.59)

In these Lagrangian problems, $\tau(u)$ and $\rho(u)$ are called the Lagrangian dual functions. The corresponding LD is obtained by maximizing $\tau(u)$ and $\rho(u)$ with respect to the multiplier vector $u \geq 0$, i.e.,

$$LD_I: \qquad D_I = \max_{u\geq 0}\left[\tau(u)\right] \quad \text{and} \quad LD_{II}: \qquad D_{II} = \max_{u\geq 0}\left[\rho(u)\right]$$

respectively. In the following theorem, the weak Lagrangian duality of LR is established. The result shows that LR is indeed a relaxation of the original (primal) problem (MIP) since its value is always a lower bound for the minimum value of MIP.

Theorem 3.5.1. *Let $v(MIP)$ be the optimal value of MIP. For all $u \in \Re^n_+$, $v(LR) \leq v(MIP)$ and $v(D) \leq v(MIP)$*

Proof.

$$
\begin{aligned}
v(MIP) &= \text{Min}\left[\sum_{(i,j)\in A} d_{ij}\phi_{ij} : (3.46-55)\&(3.57-59)\right] \\
&\geq \text{Min}\left[\sum_{(i,j)\in A} d_{ij}\phi_{ij} : \sum_{i\in N}\left(\sum_{j\in N}\phi_{ij}-1\right)=0,\ (3.47-55)\ \&\ (3.57-59)\ \text{ hold}\right] \\
&\geq \text{Min}\left[\sum_{(i,j)\in A} d_{ij}\phi_{ij} + \sum_{i\in N}\left(\sum_{j\in N}\phi_{ij}-1\right) : \sum_{i\in N}\left(\sum_{j\in N}\phi_{ij}-1\right)=0,\ (3.47-55)\ \&\ (3.57-59)\ \text{ hold}\right] \\
&\geq \text{Min}\left[\sum_{(i,j)\in A} d_{ij}\phi_{ij} + \sum_{i\in N}\left(\sum_{j\in N}\phi_{ij}-1\right) : (3.47-55)\ \&\ (3.57-59)\ \text{hold}\right] \\
&= v(LR)
\end{aligned}
$$

Thus, $v(LR) \leq v(MIP)$ for all $u \geq 0$. By implication, it is true that $v(D) \leq v(MIP)$. □

To apply the sub-gradient optimization, a sub-gradient, s, corresponding to the relaxed constraints is defined. This parameter is given by

$$
s^i = \left[\left(\sum_{j\in N}\phi_{ij}-1\right)_i\right] \tag{3.61}
$$

We adopt a simple form of the sub-gradient method method described in Held *et al.* (1974) to find the solution to the dual problems above.
A Sub-gradient Algorithm for solving LD_I and LD_{II} is as follows

Step 1: Initialize the SO parameters as follow:$u = 0$, $m = 1$ (iteration counter), $t_i : 0 < t_i \leq 2$ (the control parameter), $l = 0$ (lower bound value), $u = N$ (upper bound value).

Step 2: Solve the Lagrangian Problem with $u_i = u_i^{m-1}$. **If** $L(u_i) > l$, **then** $l = L(u_i)$

Step 3: Evaluate the sub-gradient according to (3.61).

Step 4: Compute the step size:

$$
\sigma = \frac{t(u-l)}{\|s^i\|^2}
$$

Step 5: Update the Lagrangian multipliers:

$$
u_i^m = u_i^{m-1} + \sigma_m\left[\sum_{j\in N}\phi_{ij}-1\right]
$$

Step 6: Repeat process until lower bound on the objective function is found.

3.6 Study Area

For the purpose of this research, Lagos State, a state in South West Nigeria, was considered as the study area. However, due to the complexity involved with most optimization models, major focus was on Eti-Osa Local Government Area (LGA). This LGA was chosen because it hosts two of the important districts of Lagos, that is, Victoria Island and Lekki. The former of which is particularly known for corporate business activities. The moderately good road network in this LGA made it suitable for the research reported in this thesis. Online report, Wikipedia (2013), shows that the LGA is approximately 300,000 in population. In Lagos, there are several locations of waste containers and three active landfill sites: Olushosun, Solous II and III. In addition, there are two satellite landfill sites in Ewuelepe and Epe. According to LAWMA (2014), there are about 15,000 vehicular trips to these landfill and total deposition of $298,800m^3$ of waste. Presently, the Olushosun site receives about 40 percent of the total waste generated in Lagos.

3.7 Data Description and Characteristics

This section provides a detailed description of the various data used to implement the two models developed in this thesis. The characteristics as well as the nature of the data are also discussed i two separate sections for each of the models.

3.7.1 Description and Characteristics of Data Sets for Implementing Solid Waste Collection Model

The SWC model presented in the next chapter was furnished with parameters whose data are not readily available in the literature. Hence, in order to implement the model, five data sets were randomly generated. These data correspond to basic information available on particular region in Nigeria. In particular, the Eti-Osa LGA of Lagos State was considered to obtain values for the quantities of wastes generated and the location of clusters. Three waste materials were considered: plastic, putrescibles and paper with the daily per-capita generation rate of $4kg$, $68kg$ and $14kg$ respectively in Lagos (Oguweleka (2009)). Lagos has a total of twenty LGAs and it is assumed that wastes are uniformly generated by all the inhabitants and that the population spread is uniform among the LGAs. Based on these assumptions, the per-capita rate of generation for the three wastes are $0.2kg/day$, $3.4kg/day$ and $0.7kg/day$ respectively. The various values of the rates of generation for each cluster were then generated randomly using the Minitab 17 statistical package.

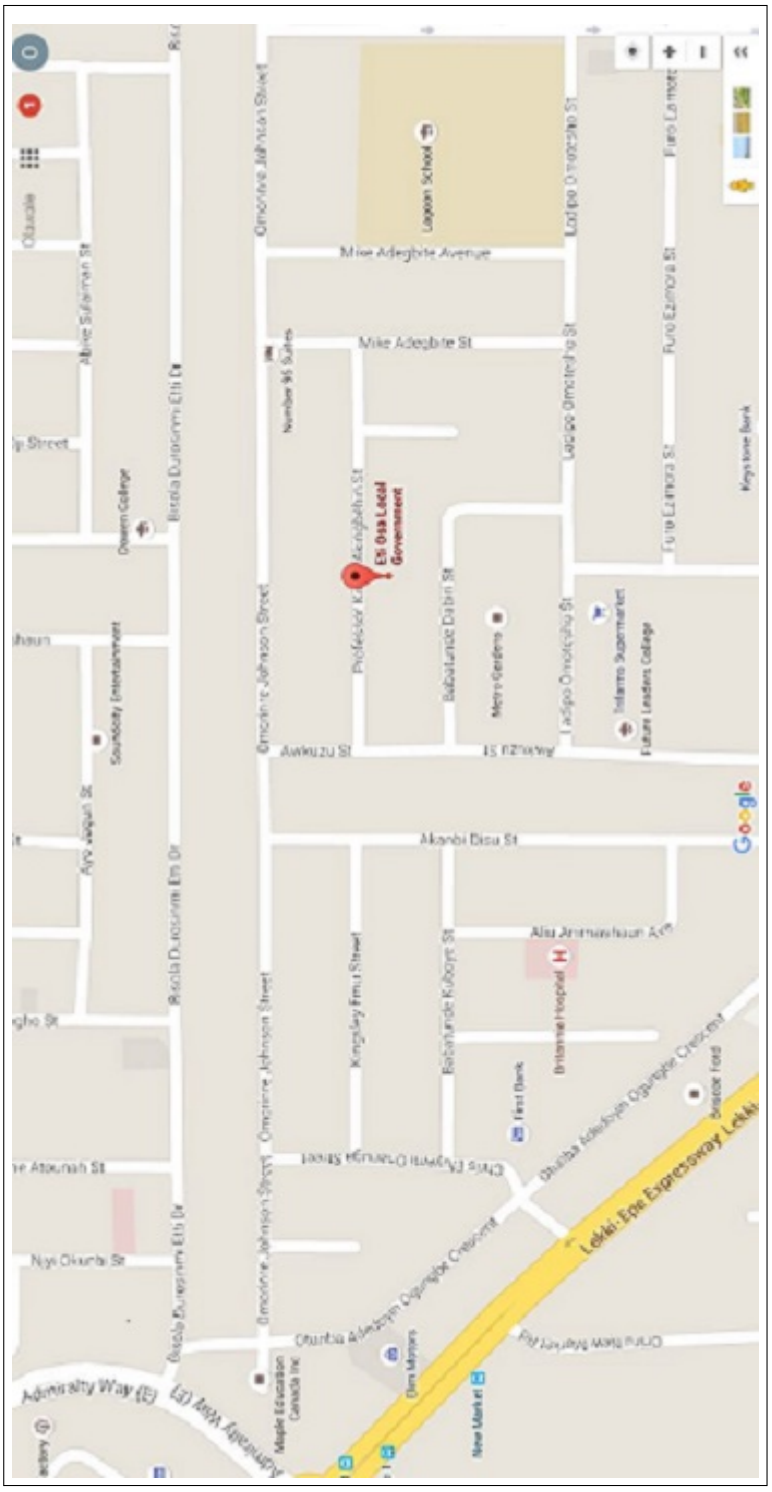

Figure 3.2: The Google Map page of Eti-Osa LGA (Adapted from the Google, 2016)

To obtain the values for the distances between the clusters and the potential collection sites, the Google map (Figure 3.2) was used to obtain the coordinate of several points (randomly) in the form (a, b, c), where a is the specific name of the location, b is the longitude of the point and c is the latitude. To accommodate several locations, the data generation approach was simplified by obtaining only the coordinates at the extreme points which define the interval of the different locations in the area. This information was then fetched into the Geographic Distance Matrix Calculator (GDMC), a java-based software, to generate the distances between the different locations (see sample screen in Figure 3.3). A full description of the data sets used for the computational experiments with SWC Model is as given in Table 3.1.

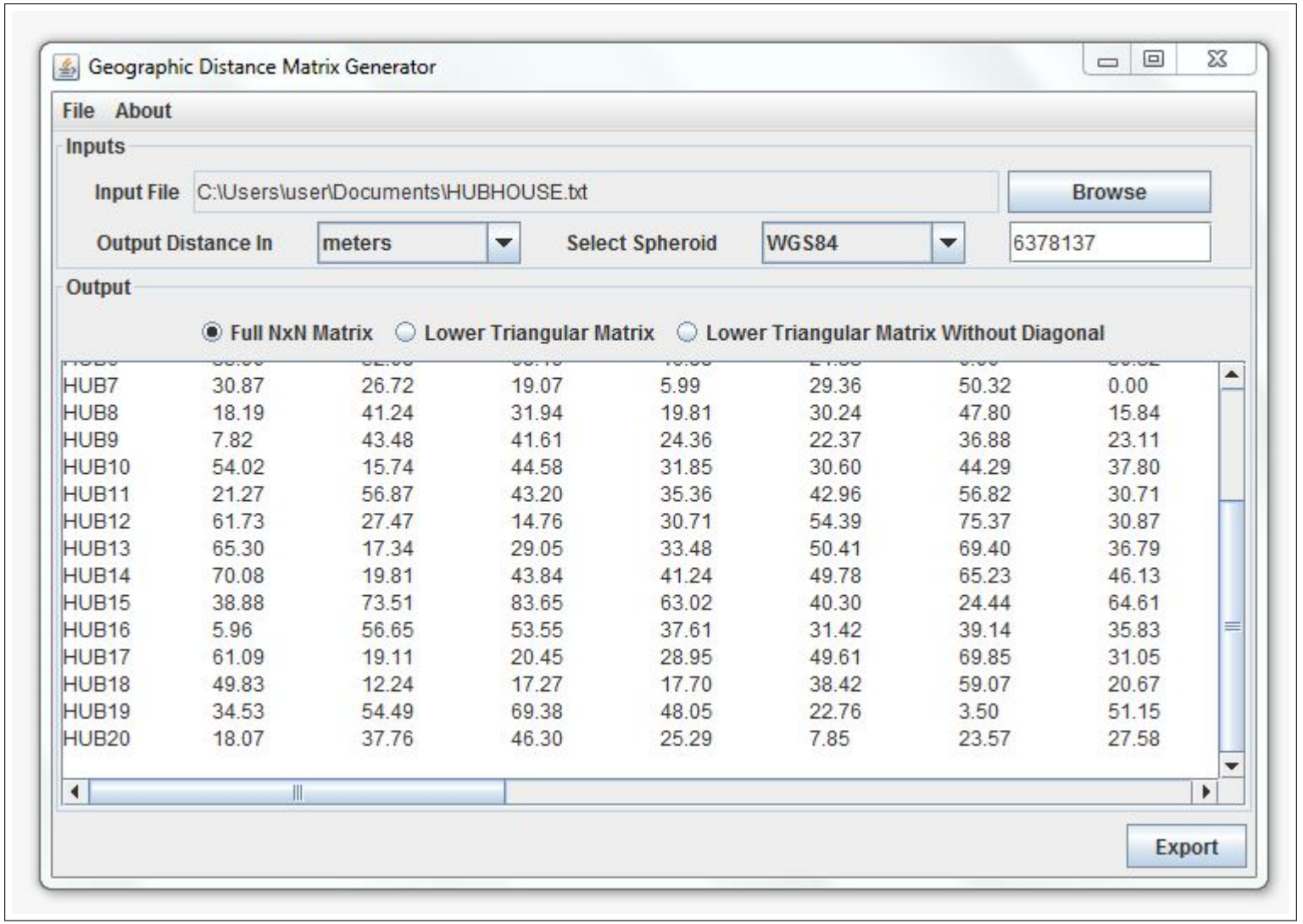

Figure 3.3: A typical GDMC screen

Table 3.1: Description of the data set for implementing SWC Model

Problem set	**Type of waste**	**Capacity of containers (kg)**	**Number of containers**	**Number of cluster**
Set 1	Plastic Putrescibles Paper	50 60 70	(21/100/45)	100
Set 2	Plastic Putrescibles Paper	50 60 70	(42/200/90)	200
Set 3	Plastic Putrescibles Paper	50 60 70	(63/300/135)	300
Set 4	Plastic Putrescibles Paper	50 60 70	(84/400/180)	400
Set 5	Plastic Putrescibles Paper	50 60 70	(105/500/225)	500

3.7.2 Description and Characteristics of Data Sets for Implementing Solid Waste Disposal Model

The numerical tests conducted on the disposal model was based on the test instances developed by Solomon (1987). There are several problem instances that have been created for the VRPTW and a collection of these can be found in the OR library (Beasley, 1990). However, the Solomon (1987) instances were preferred as the contents represent basic information needed to test the effectiveness of the model. Furthermore, many studies that have been conducted in the past and reported in literature have shown that the medium sized data of Solomon (1987) with maximum 100 nodes can adequately be utilized to test the computational effectiveness of models such as reported in this thesis.

There are six classes of data described in the Solomon (1987) test suite based on two criteria. The first criterion concerns the spatial position of customers. In this category, customers may either be in clusters (C-type), or may be randomly positioned (R-type), or a middle point may be found between clustered and randomly located customers to give the RC-type problems. The second criterion is based on the tightness of the time windows the size of the planning period. There are two categories defined for these features. When there is a tight time window and a small planning time, the problem is denoted as type 1 and when there is a large time window and planning time, the problem is denoted as type 2. Therefore, for each customer size, there are six problem instances namely **C1**, **C2**, **R1**, **R2**, **RC1** and **RC2**. For instance, **C1** test instance represents cases with clustered customers and small time windows and planning time. The Solomon test Suite is defined for three customer sizes of 25, 50 and 100. This means that there are 168 problems altogether.

CHAPTER FOUR

RESULTS

4.1 Introduction

In this chapter, the results obtained in this study are presented. In subsequent sections, results based on the computational implementations of the proposed SWC and SWD models are presented both in tabular and graphical forms. Recall that the two models were formulated as IPs. Whereas the SWC model was formulated as a set covering FLP whereby a simple partitioning of the collection area/region into clusters was performed and the potential collection sites were assumed to be a subset of the set of clusters, the SWD model handles the problem of waste disposal from the various open collection sites to the final disposal sites making use of designated vehicles. The mathematical model was modeled as a VRP which was earlier described in chapters one and two. The main contribution of the SWD model was the inclusion of the road attributes parameters. These parameters which ensure that vehicles use only roads in good condition was added to ensure that the life span of vehicles is preserved as much as possible.

4.2 Results from the Implementation of the SWC Model

Based on the study area and the data descriptions of chapter 3, this section details the results obtained from the computational experiments performed on the SWC model of section 3.3.1 and analyzed in section 3.4.

Table 4.1: Results obtained from the implementation of SWC Model with 100 clusters

Number of cluster	Threshold distance	No of potential site	Objective value	%Gap	No of iteration	Execution time	No of branch & bound nodes	Best integer solution	Best node solution	Total cuts applied
Capacity of container is 50										
100	100	100	80	-20.00	0	0.0468	0	8.00e+001	7.90e+001	0
	125		72	-28.00	0	0.0468	0	7.20e+001	7.10e+001	0
	150		60	-40.00	0	0.0936	0	6.00e+001	5.90e+001	0
	175		51	-49.00	0	0.0468	0	5.10e+001	5.02e+001	0
	200		41	-59.00	20	0.0780	0	4.10e+001	4.10e+001	13
Capacity of container is 60										
100	100	100	80	-20.00	0	0.0780	0	8.00e+001	7.90e+001	0
	125		72	-28.00	0	0.0936	0	7.20e+001	7.10e+001	0
	150		60	-40.00	0	0.0936	0	6.00e+001	5.90e+001	0
	175		51	-49.00	0	0.0624	0	5.10e+001	5.02e+001	0
	200		41	-59.00	20	0.1092	0	4.10e+001	4.10e+001	13
Capacity of container is 70										
100	100	100	80	-20.00	0	0.0624	0	8.00e+001	7.90e+001	0
	125		72	-28.00	0	0.0780	0	7.20e+001	7.10e+001	0
	150		60	-40.00	0	0.1092	0	6.00e+001	5.90e+001	0
	175		51	-49.00	0	0.0780	0	5.10e+001	5.02e+001	0
	200		41	-59.00	20	0.0936	0	4.10e+001	4.10e+001	13

Table 4.2: Results obtained from the implementation of SWC Model with 200 clusters

Number of cluster	Threshold distance	No of potential site	Objective value	%Gap	No of iteration	Execution time	No of branch & bound nodes	Best integer solution	Best node solution	Total cuts applied
Capacity of container is 50										
200	100	200	88	-56.00	49	0.2028	0	8.80e+001	8.73e+001	33
	125		60	-70.00	131	0.2964	0	6.00e+001	5.60e+001	49
	150		45	-77.50	2599	1.1700	0	4.50e+001	4.50e+001	987
	175		35	-82.5	9672	4.2120	0	3.50e+001	1.00e+000	947
	200		27	-86.5	5697	4.9296	0	2.70e+001	2.65e+001	1243
Capacity of container is 60										
200	100	200	88	-56.00	49	0.2496	0	8.80e+001	8.70e+001	33
	125		60	-70.00	131	0.3120	0	6.00e+001	5.60e+001	49
	150		45	-77.50	2599	1.2012	0	4.50e+001	4.45e+001	987
	175		35	-82.5	4576	2.8080	45	3.50e+001	3.40e+001	1103
	200		27	-86.5	4717	3.0732	0	2.70e+001	2.63e+001	1257
Capacity of container is 70										
200	100	200	88	-56.00	49	0.1872	0	8.80e+001	8.70e+001	33
	125		60	-70.00	131	0.2808	0	6.00e+001	5.60e+001	49
	150		45	-77.50	2599	1.2480	0	4.50e+001	4.45e+001	987
	175		35	-82.5	4576	2.7456	45	3.50e+001	3.40e+001	1103
	200		27	-86.5	5062	3.9624	0	2.70e+001	2.63e+001	1371

Table 4.3: Results obtained from the implementation of SWC Model with 300 clusters

Number of cluster	Threshold distance	No of potential site	Objective value	%Gap	No of iteration	Execution time	No of branch & bound nodes	Best integer solution	Best node solution	Total cuts applied
Capacity of container is 50										
300	100	300	89	-70.33	2610	1.2168	0	8.90e+001	8.90e+001	1162
	125		64	-78.67	4738	3.2916	0	6.40e+001	6.35e+001	1598
	150		47	-85.33	9585	8.2524	0	4.70e+001	4.70e+001	1206
	175		36	-88.00	394793	343.5296	2887	3.60e+001	3.40e+001	3525
	200		29	-90.33	31006	28.4234	241	2.90e+001	2.87e+001	2222
Capacity of container is 60										
300	100	300	89	-70.33	2610	1.0764	0	8.90e+001	8.90e+001	1162
	125		64	-78.67	4738	3.1668	0	6.40e+001	6.35e+001	1598
	150		47	-85.33	9177	8.0964	19	4.70e+001	4.70e+001	1808
	175		36	-88.00	553664	554.4280	2736	3.60e+001	3.40e+001	4155
	200		29	-90.33	141377	142.9440	2745	2.90e+001	2.90e+001	2896
Capacity of container is 70										
300	100	300	89	-70.33	2610	1.1076	0	8.90e+001	8.90e+001	1162
	125		64	-78.67	4738	3.2448	0	6.40e+001	6.35e+001	1598
	150		47	-85.33	7721	5.7720	0	4.70e+001	4.68e+001	1817
	175		36	-88.00	972413	798.5066	6294	3.60e+001	3.40e+001	4516
	200		29	-90.33	62809	37.3934	825	2.90e+001	2.80e+001	2529

Table 4.4: Results obtained from the implementation of SWC Model with 400 clusters

Number of cluster	Threshold distance	No of potential site	Objective value	%Gap	No of iteration	Execution time	No of branch & bound nodes	Best integer solution	Best node solution	Total cuts applied
Capacity of container is 50										
400	100	400	99	-75.25	4067	1.9344	0	9.90e+001	9.90e+001	1671
	125		67	-83.25	7348	7.1916	0	6.70e+001	6.63e+001	2097
	150		46	-88.50	34225	27.2690	476	4.60e+001	4.51e+001	2683
	175		36	-91.00	2727039	3563.9512	10799	3.60e+001	3.50e+001	6010
	200		28	-93.00	3430549	6186.2156	21373	2.80e+001	2.80e+001	5811
Capacity of container is 60										
400	100	400	99	-75.25	4067	1.9344	0	9.90e+001	9.90e+001	1671
	125		67	-83.25	8396	7.7532	0	6.70e+001	6.63e+001	2193
	150		46	-88.50	37015	29.8274	394	4.60e+001	4.53e+001	2775
	175		36	-91.00	608737	749.5696	6232	3.60e+001	3.55e+001	5017
	200		28	-93.00	1072422	1569.6780	10037	2.80e+001	2.80e+001	5247
Capacity of container is 70										
400	100	400	99	-75.25	4067	2.0748	0	9.90e+001	9.90e+001	1671
	125		67	-83.25	7385	7.1604	20	6.70e+001	6.65e+001	2266
	150		46	-88.50	14977	21.8713	0	4.60e+001	4.55e+001	2926
	175		36	-91.00	949541	1475.3800	11226	3.60e+001	3.57e+001	5312
	200		28	-93.00	5398776	7636.8612	14955	2.80e+001	2.75e+001	6022

Table 4.5: Results obtained from the implementation of SWC Model with 500 clusters

Capacity of container is 50										
Number of cluster	Threshold distance	No of potential site	Objective value	%Gap	No of iteration	Execution time	No of branch & bound nodes	Best integer solution	Best node solution	Total cuts applied
500	100	500	95	-81.00	7432	5.8344	0	9.50e+001	9.40e+001	2521
	125		66	-86.80	102077	136.8596	1613	6.60e+001	6.60e+001	3523
	150		48	-90.40	2863615	4793.2524	9743	4.80e+001	4.70e+001	7492
Capacity of container is 60										
500	100	500	95	-81.00	9087	8.2056	34	9.50e+001	9.44e+001	2455
	125		66	-86.80	103479	141.2436	1957	6.60e+001	6.60e+001	3844
	150		48	-90.40	915782	1458.4556	3229	4.80e+001	4.80e+001	6827
Capacity of container is 70										
500	100	500	95	-81.00	9087	7.8780	39	9.50e+001	9.44e+001	2456
	125		66	-86.80	130479	143.412	1957	6.60e+001	6.60e+001	3844
	150		48	-90.40	754111	1346.0256	3963	4.80e+001	4.75e+001	6200

Table 4.6: Results obtained for container allocation for $c = 50$

No of Clusters	D	Current Allocation			Total	New Allocation			Total	% Reduction
		t1	t2	t2		t1	t2	t3		
100	150	21	100	45	166	0	60	0	60	-63.85
	200	21	100	45	166	0	41	0	41	-75.30
200	150	42	200	90	332	5	38	4	47	-85.84
	200	42	200	90	332	10	13	9	32	-90.36
300	150	63	300	135	498	4	38	7	49	-90.16
	200	63	300	135	498	16	10	12	38	-92.37
400	150	84	400	180	664	15	21	18	54	-91.87
	200	84	400	180	664	14	20	18	52	-92.17
500	150	105	500	225	830	37	11	30	78	-90.60
	200	105	500	225	830	-	-	-	-	-

Table 4.7: Results obtained for container allocation for $c = 60$

No of Clusters	D	Current Allocation			Total	New Allocation			Total	% Reduction
		t1	t2	t2		t1	t2	t3		
100	150	21	100	45	166	0	60	0	60	-63.85
	200	21	100	45	166	0	41	0	41	-75.30
200	150	42	200	90	332	5	38	4	47	-85.84
	200	42	200	90	332	12	8	8	28	-91.57
300	150	63	300	135	498	15	14	18	47	-90.56
	200	63	300	135	498	12	7	13	32	-93.57
400	150	84	400	180	664	12	20	15	47	-92.92
	200	84	400	180	664	11	16	12	39	-94.13
500	150	105	500	225	830	10	19	29	58	-93.01
	200	105	500	225	830	-	-	-	-	-

Table 4.8: Results obtained for container allocation for $c = 70$

No of Clusters	D	Current Allocation			Total	New Allocation			Total	% Reduction
		t1	t2	t2		t1	t2	t3		
100	150	21	100	45	166	0	60	0	60	-63.85
	200	21	100	45	166	0	41	0	41	-75.30
200	150	42	200	90	332	5	38	4	47	-85.84
	200	42	200	90	332	10	10	8	28	-91.57
300	150	63	300	135	498	11	34	8	53	-89.36
	200	63	300	135	498	9	14	12	35	-92.97
400	150	84	400	180	664	15	18	13	46	-93.07
	200	84	400	180	664	19	25	6	50	-92.47
500	150	105	500	225	830	14	21	19	54	-93.49
	200	105	500	225	830	-	-	-	-	-

Table 4.9: Applied cuts in Model 1 for $c = 50$

Number of Clusters	Threshold distance	Types of Cuts					
		Clique	Cover	Flow	Mixed Inter Rounding	Zero-half	Gomory Fractional
100	150	0	0	0	0	0	0
	200	3	8	0	0	1	1
200	150	548	234	2	90	70	43
	200	757	245	16	142	46	21
300	150	745	188	34	148	31	54
	200	1209	470	50	402	78	13
400	150	1541	535	59	329	118	24
	200	1926	2274	39	711	202	15
500	150	3030	2748	66	893	362	37
	200	-	-	-	-	-	-

Table 4.10: Applied cuts in Model 1 for $c = 60$

Number of Clusters	Threshold distance	Types of Cuts					
		Clique	Cover	Flow	Mixed Inter Rounding	Zero-half	Gomory Fractional
100	150	0	0	0	0	0	0
	200	3	8	0	0	1	1
200	150	548	234	2	90	70	43
	200	719	259	20	161	61	28
300	150	1084	293	39	250	114	28
	200	1467	752	45	437	130	17
400	150	1589	571	71	388	92	32
	200	2018	1797	68	744	308	13
500	150	3117	2306	130	1024	223	27
	200	-	-	-	-	-	-

Table 4.11: Applied cuts in Model 1 for $c = 70$

Number of Clusters	Threshold distance	Types of Cuts					
		Clique	Cover	Flow	Mixed Inter Rounding	Zero-half	Gomory Fractional
100	150	0	0	0	0	0	0
	200	3	8	0	0	1	1
200	150	548	234	2	90	70	43
	200	861	260	21	151	50	25
300	150	1056	323	42	234	113	48
	200	1478	434	63	441	87	8
400	150	1916	441	49	364	109	37
	200	2333	2017	82	814	506	28
500	150	3410	1438	97	913	301	33
	200	-	-	-	-	-	-

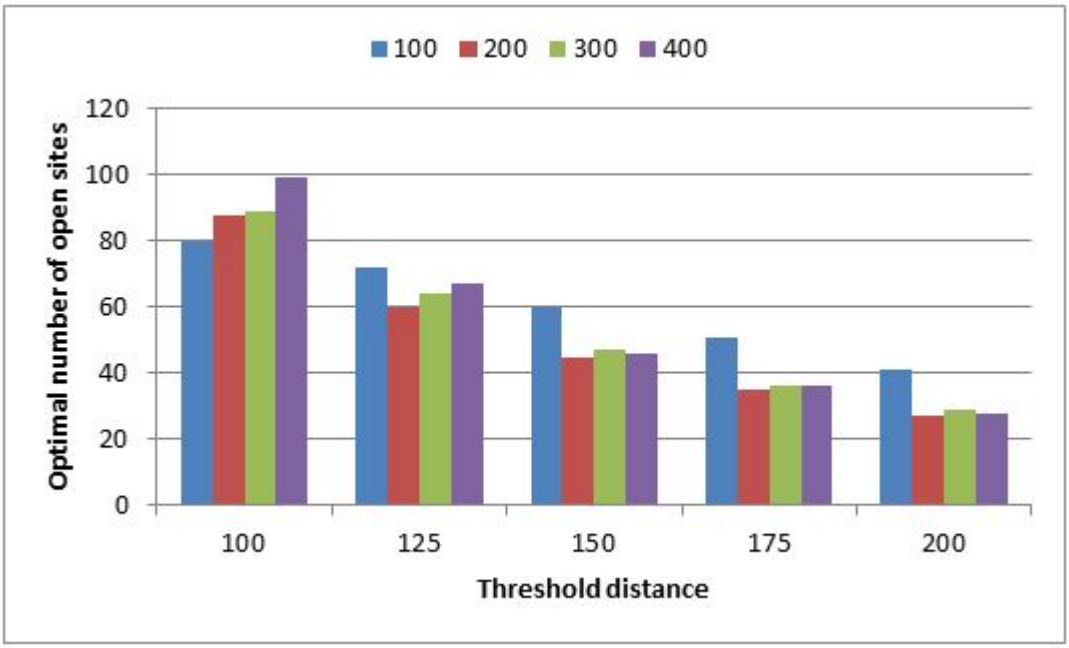

Figure 4.1: Number of open waste collection sites for each threshold distance

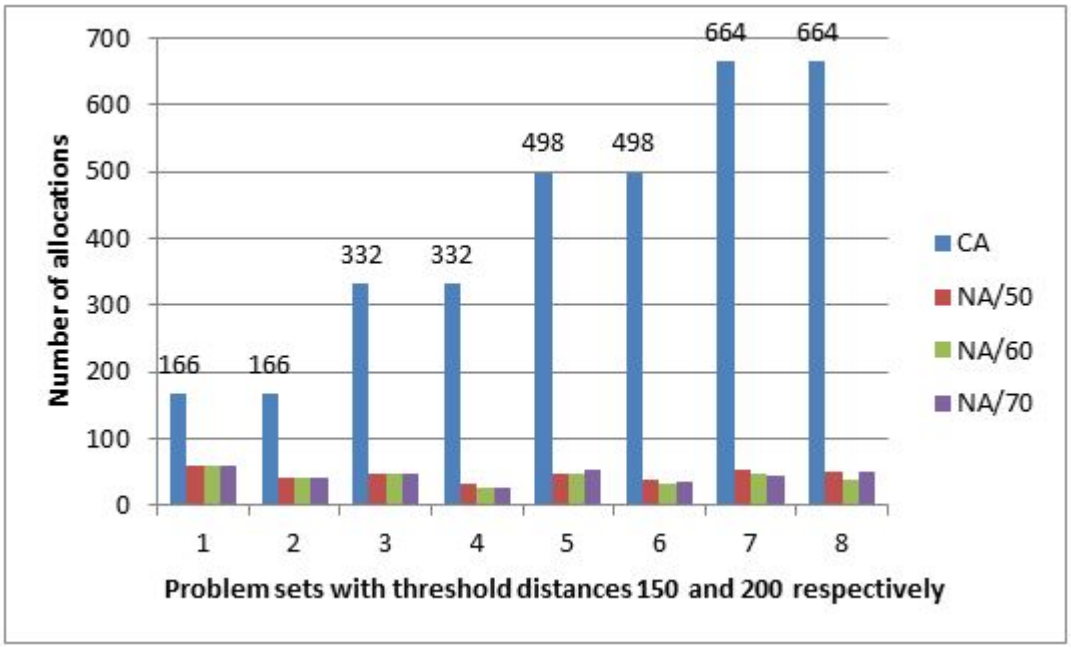

Figure 4.2: Graphical illustration of new container allocation vs. current allocation

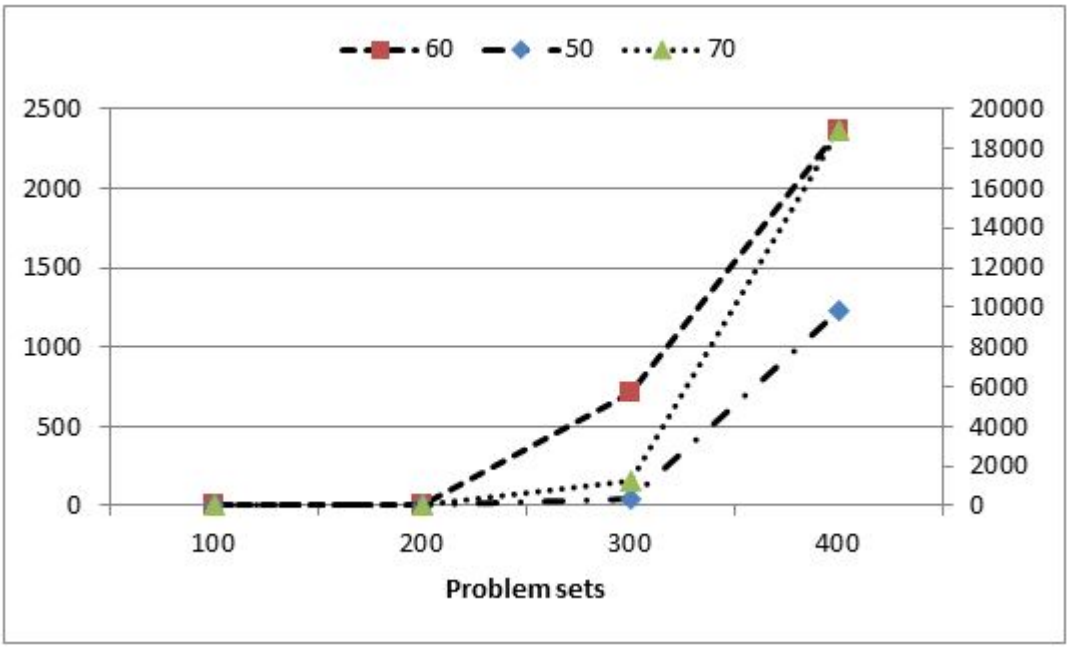

Figure 4.3: The total execution time for four problem sets

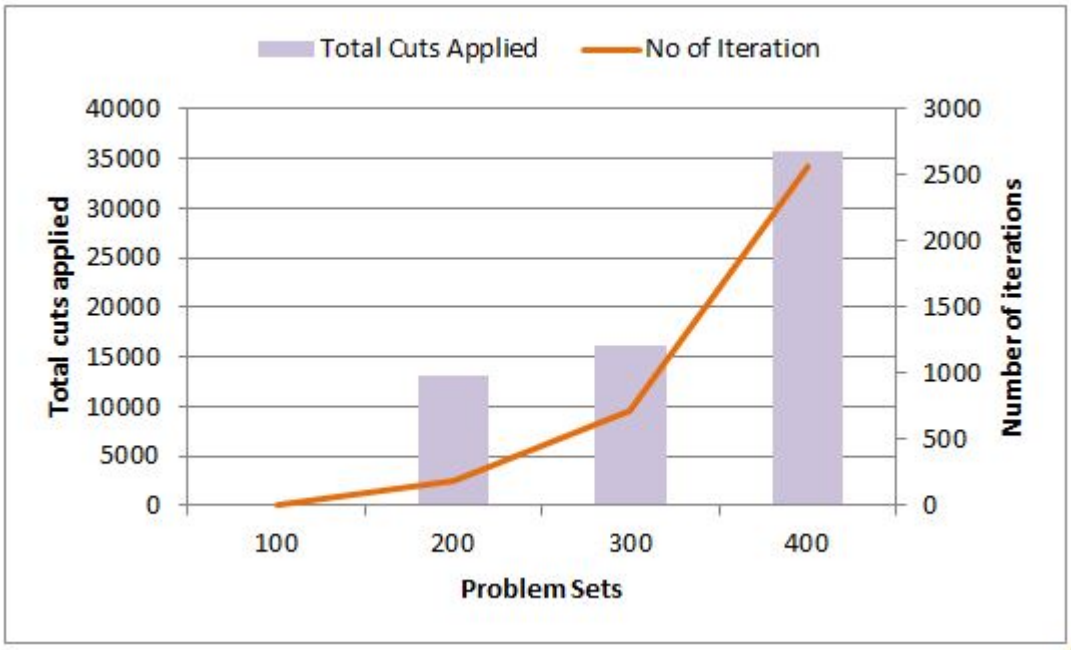

Figure 4.4: Total cuts applied vs. number of iterations at threshold distance of 150

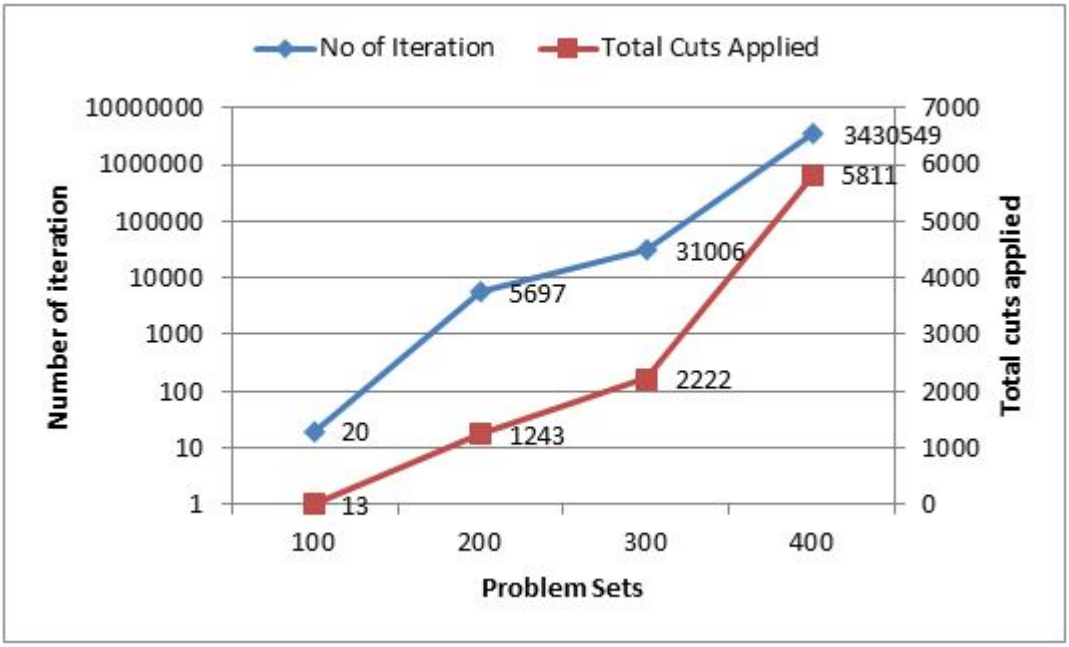

Figure 4.5: Total cuts applied vs. number of iterations at threshold distance of 200

4.3 Results from the Implementation of SWD Model

In this section, the numerical results from the computational test on the SWD model with the Solomon (1987) instances described in chapter three are presented. As earlier illustrated, there were two cases to the relaxed problem: one in which the objective function was linear and the other where the objective function was nonlinear. These two cases were tested and their results were compared. The results were also benchmarked with published results on single VRPTW. Tables 4.12-4.19 give the results of these computations.

Table 4.12: Result for Case I with 25 customers and $R \geq 0.4$

Test Instance	NLO	TD	Iter	Exec Time	AA	NSC	SSR		
							Opt	ExIter	CPNI
R101	936	146.57	1392	1.030	51	2			√
R102	1518	132.03	2375	2.090	46	4	√		
R103	1125	131.96	1643	1.263	42	4	√		
R104	992	131.72	1535	1.217	42	5	√		
R105	993	132.58	1102	1.045	49	3			√
R106	1530	131.74	1677	1.654	49	4	√		
R107	1318	131.65	2057	1.669	43	5	√		
R108	922	111.56	1209	1.186	45	4	√		
R109	902	101.33	1115	3.073	41	3	√		
R110	1672	131.83	2156	1.981	42	7	√		
R111	902	131.93	1271	1.763	41	5	√		
R112	1000	121.12	1410	1.435	48	1		√	
R201	952	149.16	1323	1.217	47	2	√		
R202	967	155.03	1384	1.108	43	3	√		
R203	736	154.21	1185	1.248	44	4	√		
R204	1215	119.38	1982	1.747	39	3	√		
R205	860	171.440	1409	2.246	49	2			√
R206	751	153.86	1122	0.967	45	5	√		
R207	1060	153.19	1542	4.024	43	3	√		
R208	838	98.97	1438	1.279	40	2	√		
R209	974	146.00	1563	1.139	44	4	√		
R210	791	123.27	1361	2.028	45	3	√		
R211	1035	133.63	1532	1.280	54	2	√		
RC101	1425	74.60	2293	5.975	46	5	√		
RC102	914	64.76	1519	2.199	42	7			√
RC103	887	75.55	1378	1.997	49	8			√
RC104	732	58.09	1359	2.262	45	5	√		
RC105	667	74.69	1032	0.936	44	6	√		
RC106	1103	74.44	1762	1.466	47	5			√
RC107	605	74.42	1172	1.684	45	7			√
RC108	986	74.31	1579	1.170	45	7	√		
RC201	1241	74.74	1921	1.389	64	2		√	
RC202	716	76.39	1048	0.874	44	3	√		
RC203	605	58.96	1133	0.827	44	4	√		
RC204	1036	59.14	1578	1.264	44	7	√		
RC205	1019	58.21	1748	1.466	41	5		√	
RC206	731	39.90	1303	1.731	38	3	√		
RC207	1251	65.82	1881	1.763	44	5			√
RC208	736	74.42	1247	1.014	44	7	√		

Table 4.13: Result for Case I with 25 customers and $R \geq 0.5$

Test Instance	NLO	TD	Iter	Exec Time	AA	NSC	SSR		
							Opt	ExIter	CPNI
C102	1023	64.67	1667	1.732	54	4	√		
C104	1123	62.04	1628	1.311	44	5	√		
C105	860	80.66	1548	1.279	54	1	√		
C107	1144	88.12	1953	1.373	60	2		√	
C108	1517	73.48	1757	1.872	52	2	√		
C109	1221	73.38	1726	1.466	58	2		√	
R101	1502	169.78	1975	1.529	49	3	√		
R102	1129	170.14	1508	1.357	45	5		√	
R103	1412	200.84	1406	1.216	46	7	√		
R104	1066	170.19	1716	1.388	46	6	√		
R105	1103	195.22	1465	1.326	52	5	√		
R106	1219	182.00	1401	1.482	47	6	√		
R107	1984	170.20	2315	2.496	45	6		√	
R108	1894	178.53	1894	1.497	55	4		√	
R109	640	195.07	1057	0.936	45	6	√		
R110	1397	194.11	1952	1.840	47	9		√	
R111	1355	171.33	1974	1.716	49	5		√	
R112	844	195.01	1243	1.295	45	6	√		
R201	1067	197.22	1722	2.465	47	6	√		
R202	1215	197.14	1884	1.966	46	6	√		
R203	1092	199.37	1787	1.482	51	6	√		
R204	1389	201.61	2224	1.482	45	7		√	
R205	1016	174.62	1596	1.372	48	8	√		
R206	1387	195.79	2279	1.685	47	7		√	
R207	1105	195.58	1676	1.373	45	5	√		
R208	903	194.25	1427	1.061	45	7	√		
R209	646	228.44	1147	0.795	56	5			√
R210	2172	196.06	2055	1.794	48	6	√		
R211	1288	221.60	1772	1.263	54	3		√	
RC101	597	88.80	1112	6.817	46	9	√		
RC102	917	88.73	1281	2.059	43	12	√		
RC103	262	326.57	409	1.669	57	1			√
RC104	999	79.79	1739	1.467	42	10	√		
RC105	792	88.55	1370	2.496	47	12	√		
RC106	457	76.63	842	0.733	41	8	√		
RC107	588	88.33	1191	0.811	47	9	√		
RC108	1073	34.76	1659	1.310	45	5	√		
RC201	668	80.59	1089	0.873	45	7	√		
RC202	782	89.32	1326	0.967	40	6			√
RC203	998	77.43	1679	1.170	40	7	√		
RC204	791	77.46	1303	0.921	44	7	√		
RC205	1417	117.47	2511	1.575	55	5		√	
RC206	1227	89.88	1839	1.264	44	6	√		
RC207	879	62.94	1427	1.060	39	7	√		
RC208	1443	91.46	2154	1.435	42	10		√	

Table 4.14: Result for Case I with 50 customers and $R \geq 0.4$

Test Instance	NLO	TD	Iter	Exec Time	AA	NSC	SSR		
							Opt	ExIter	CPNI
R101	1278	396.57	2530	7.722	131	2		√	
R102	1853	447.21	2829	10.437	115	7		√	
R103	1670	413.60	2594	8.205	112	7		√	
R104	1307	471.28	3090	8.908	110	9		√	
R105	1187	473.33	2660	6.615	108	6		√	
R106	1345	378.98	2496	7.379	110	7		√	
R107	1301	410.63	2691	7.598	112	7		√	
R108	1501	373.07	2720	8.112	111	5		√	
R109	1669	343.85	2712	9.516	108	6		√	
R110	1337	419.93	2381	6.724	112	5		√	
R111	1367	371.62	2396	7.488	104	6		√	
R112	1233	338.51	2274	6.817	104	6		√	
R201	924	530.12	1390	4.431	115	7		√	
R202	1220	379.72	2647	7.051	120	7		√	
R203	1925	365.72	2682	10.936	113	6		√	
R204	1258	406.70	2691	6.864	114	6		√	
R205	1358	441.39	2459	7.004	118	6		√	
R206	1267	374.44	2489	8.455	107	3		√	
R207	1340	400.46	2515	7.706	122	3		√	
R208	1800	515.67	2797	8.673	116	10		√	
R209	1413	355.58	2358	12.574	106	4		√	
R210	1302	324.67	2466	8.050	143	4		√	
R211	1203	435.24	2458	8.377	120	6		√	
RC101	1311	421.65	2475	6.833	111	5		√	
RC102	934	562.55	2737	5.881	112	4		√	
RC103	1048	245.44	2710	6.568	103	5		√	
RC104	1019	506.49	2456	5.944	142	9		√	
RC105	1202	373.56	2464	6.365	108	6		√	
RC106	2091	303.58	2536	9.813	108	4		√	
RC107	1710	280.45	2826	7.972	108	5		√	
RC108	920	365.35	2379	7.176	101	4		√	
RC201	1176	245.30	2480	6.193	109	6		√	
RC202	995	285.82	2637	6.365	113	8		√	
RC203	919	233.83	2525	5.445	108	5		√	
RC204	1972	376.16	2599	8.517	112	8		√	
RC205	1096	277.06	2347	5.803	108	5		√	
RC206	1097	298.41	2386	6.022	112	5		√	
RC207	966	173.60	2419	5.772	107	6		√	
RC208	1424	338.17	2417	7.722	112	6		√	

Table 4.15: Result for Case I with 50 customers and $R \geq 0.5$

Test Instance	NLO	TD	Iter	Exec Time	AA	NSC	SSR		
							Opt	ExIter	CPNI
R101	1232	523.25	2396	6.598	128	9		√	
R102	1509	770.47	2579	9.438	154	5		√	
R103	1326	694.87	2392	8.455	117	10		√	
R104	1353	816.47	2435	8.035	114	11		√	
R105	1244	588.30	2561	8.627	123	11		√	
R106	1155	843.57	1914	6.147	131	2			√
R107	1347	801.13	2406	8.627	134	10		√	
R108	1105	789.25	2528	6.676	113	9		√	
R109	2233	492.29	2598	12.308	129	9		√	
R110	1287	587.67	2519	7.472	119	9		√	
R111	1309	712.23	2489	8.658	135	10		√	
R112	1347	836.91	3017	7.909	122	11		√	
R201	1449	503.43	2495	7.316	130	9		√	
R202	1175	582.74	2438	6.489	121	9		√	
R203	1616	739.39	2559	7.894	130	7		√	
R204	1708	758.04	2520	8.128	124	6		√	
R205	1604	515.41	2494	7.909	131	12		√	
R206	2166	779.21	2498	9.111	130	7		√	
R207	1230	714.90	2304	6.489	118	7		√	
R208	1326	669.02	2440	6.864	127	6		√	
R209	2301	577.95	2595	9.891	129	12		√	
R210	1617	610.96	2364	7.738	139	5		√	
R211	1467	507.92	2316	7.270	125	8		√	
RC101	856	946.84	1318	4.197	120	5			√
RC102	1566	548.83	2471	8.299	114	7		√	
RC103	1128	485.38	2616	6.599	130	8		√	
RC104	1088	731.42	2659	6.271	109	9		√	
RC105	1673	382.84	2589	7.956	111	6		√	
RC106	989	441.40	2459	5.819	117	5		√	
RC107	814	971.27	1155	3.853	121	-			√
RC108	1317	396.66	2455	6.817	115	10		√	
RC201	1094	547.01	2371	6.084	125	9		√	
RC202	1032	510.74	2569	6.147	124	9		√	
RC203	1595	440.55	2619	8.252	129	8		√	
RC204	1070	651.97	2448	6.068	115	6		√	
RC205	969	484.65	2359	5.912	117	8		√	
RC206	1896	460.16	2560	8.627	130	7		√	
RC207	1489	624.47	2497	7.488	123	9		√	
RC208	1558	669.42	2400	7.566	127	7		√	

Table 4.16: Result for Case II with 25 customers and $R \geq 0.4$

Test Instance	TD	Iter	Exec Time	AA	NSC	SSR		
						Opt	ExIter	CPNI
R101	132.68	738	1.887	47	4	√		
R102	153.99	835	0.624	46	4			√
R103	111.49	936	0.764	43	4	√		
R104	112.02	1181	0.827	48	4	√		
R105	101.45	936	0.718	47	3	√		
R106	101.06	844	0.640	45	2	√		
R107	98.84	979	0.749	46	5	√		
R108	91.72	941	0.764	40	5	√		
R109	90.56	944	0.733	44	4	√		
R110	61.05	931	0.671	42	1	√		
R111	131.95	1045	0.874	47	7	√		
R112	71.11	1289	0.983	42	4	√		
R201	135.57	909	0.656	43	3	√		
R202	46.26	1150	0.936	47	1	√		
R203	116.91	1490	1.123	48	4	√		
R204	119.32	1164	0.796	42	4	√		
R205	114.14	815	0.811	42	3	√		
R206	28.58	1039	0.905	42	2	√		
R207	39.50	1203	0.858	46	2	√		
R208	26.16	1242	0.936	43	1	√		
R209	104.72	1031	0.874	44	2	√		
R210	154.04	823	0.717	46	4	√		
R211	6.63	833	0.733	41	1	√		
RC101	57.88	728	0.608	44	2	√		
RC102	74.58	790	0.624	45	7	√		
RC103	74.53	726	0.702	45	7	√		
RC104	33.60	1075	0.796	43	5	√		
RC105	74.69	709	0.531	44	6	√		
RC106	65.01	699	0.561	45	4	√		
RC107	39.30	848	0.702	48	6	√		
RC108	59.60	814	0.686	46	5			√
RC201	76.56	701	0.640	45	5	√		
RC202	76.39	678	0.593	44	3	√		
RC203	44.14	1178	0.873	45	4	√		
RC204	18.71	1352	0.983	44	6	√		
RC205	76.42	1494	1.046	45	6	√		
RC206	43.79	958	0.718	42	4	√		
RC207	39.53	998	0.702	48	4	√		
RC208	13.03	1518	1.638	38	5	√		

Table 4.17: Result for Case II with 25 customers and $R \geq 0.5$

Test Instance	TD	Iter	Exec Time	AA	NSC	SSR		
						Opt	ExIter	CPNI
R101	195.42	751	0.702	52	3	√		
R102	170.64	1059	0.920	50	5	√		
R103	114.17	2021	1.295	48	6	√		
R104	160.60	1965	1.233	49	8	√		
R105	147.54	976	0.780	50	4	√		
R106	212.52	645	0.639	59	2	√		
R107	249.66	666	0.592	51	2	√		
R108	170.35	1178	0.921	47	6	√		
R109	208.01	1026	0.811	49	5	√		
R110	194.16	931	0.702	45	7	√		
R111	195.08	910	0.827	48	5	√		
R112	195.01	1036	0.983	48	5	√		
R201	126.99	1045	1.357	50	5	√		
R202	51.42	1918	1.763	42	4	√		
R203	170.90	1489	1.248	46	5	√		
R204	70.43	1813	1.856	43	4	√		
R205	142.49	1502	1.419	45	8	√		
R206	118.53	1470	1.436	50	4	√		
R207	195.58	1279	0.968	47	5	√		
R208	194.16	1185	1.373	49	5	√		
R209	194.21	1157	0.936	48	10	√		
R210	195.95	1369	1.264	49	8	√		
R211	163.04	1642	1.654	43	8	√		
RC101	88.80	685	0.530	49	9	√		
RC102	88.73	850	0.624	46	13	√		
RC103	76.50	747	0.578	43	7	√		
RC104	43.37	1354	0.921	43	7	√		
RC105	64.97	1143	0.749	45	6	√		
RC106	76.04	778	0.796	47	5	√		
RC107	88.08	1094	0.796	48	7	√		
RC108	88.26	1072	0.734	45	7	√		
RC201	80.59	905	0.717	48	8	√		
RC202	91.95	851	0.702	42	8	√		
RC203	26.87	1238	1.154	43	5	√		
RC204	76.22	810	0.640	42	7	√		
RC205	90.84	771	0.687	45	8	√		
RC206	89.44	926	0.921	48	7	√		
RC207	79.26	1149	0.827	40	6	√		
RC208	33.24	1350	0.905	41	8	√		

Table 4.18: Result for Case II with 50 customers and $R \geq 0.4$

Test Instance	TD	Iter	Exec Time	AA	NSC	SSR		
						Opt	ExIter	CPNI
R101	290.23	2388	6.131	93	7	√		
R102	364.60	2507	7.862	112	7		√	
R103	344.22	2494	6.146	106	8		√	
R104	548.39	1265	3.807	113	2			√
R105	289.62	2255	5.507	93	7	√		
R106	298.23	2440	5.990	104	6		√	
R107	352.79	2287	8.065	108	7		√	
R108	393.71	2667	8.315	122	10		√	
R109	299.58	2568	7.488	102	7		√	
R110	308.29	2348	6.022	103	9		√	
R111	349.23	2755	8.190	112	4		√	
R112	310.99	2546	10.998	103	7		√	
R201	307.42	2572	8.751	106	7		√	
R202	339.41	2641	8.549	104	6		√	
R203	340.52	2583	6.505	112	9		√	
R204	369.49	2647	6.926	121	9		√	
R205	305.24	2591	8.580	109	9		√	
R206	371.28	2356	9.204	114	6		√	
R207	402.89	2651	6.505	115	5		√	
R208	380.67	2656	6.817	115	7		√	
R209	283.61	2427	8.876	110	5		√	
R210	310.01	2308	9.532	103	6		√	
R211	319.85	2415	8.409	122	6		√	
RC101	107.11	1589	5.803	98	8	√		
RC102	133.48	2511	11.326	114	9		√	
RC103	122.60	2696	9.141	113	9		√	
RC104	394.59	2800	11.123	117	6		√	
RC105	85.53	2282	6.349	90	11	√		
RC106	122.77	2477	8.705	107	10		√	
RC107	110.43	2227	9.048	100	13	√		
RC108	121.12	2454	8.830	105	11		√	
RC201	112.80	2283	9.204	95	11	√		
RC202	116.20	2208	7.035	93	12			√
RC203	114.93	2452	9.064	89	11		√	
RC204	264.02	2417	8.564	131	9		√	
RC205	221.96	2316	9.500	116	3		√	
RC206	119.51	2398	8.440	101	12		√	
RC207	99.67	2428	6.833	93	12	√		
RC208	108.64	2408	7.971	92	12		√	

Table 4.19: Result for Case II with 50 customers and $R \geq 0.5$

Test Instance	TD	Iter	Exec Time	AA	NSC	SSR		
						Opt	ExIter	CPNI
R101	408.91	2281	8.533	117	11		√	
R102	600.73	2488	6.224	116	11		√	
R103	649.73	2590	6.192	119	11		√	
R104	565.69	2585	6.442	113	12		√	
R105	492.74	2601	8.205	113	12		√	
R106	533.01	2359	6.365	110	13		√	
R107	686.76	2406	6.006	115	8		√	
R108	578.22	2449	7.316	114	9		√	
R109	429.58	2399	8.252	122	12		√	
R110	484.22	2315	5.991	115	12		√	
R111	630.93	2414	6.662	120	10		√	
R112	778.11	2614	8.393	126	12		√	
R201	495.17	2220	8.799	137	9		√	
R202	478.12	2301	9.235	136	12		√	
R203	742.88	2283	7.769	140	5		√	
R204	579.16	2403	9.141	139	5		√	
R205	493.80	2482	13.619	129	11		√	
R206	621.33	2337	7.550	133	9		√	
R207	740.14	2327	9.001	134	5		√	
R208	804.25	1643	6.848	135	4		√	
R209	505.63	2623	8.253	125	10		√	
R210	455.19	2618	8.611	125	8		√	
R211	524.31	2330	6.973	134	9		√	
RC101	250.33	2475	7.832	132	8		√	
RC102	293.22	2481	8.814	133	10		√	
RC103	380.50	2561	8.003	125	8		√	
RC104	611.34	2431	8.143	124	6		√	
RC105	189.05	2395	8.128	116	15		√	
RC106	245.34	2427	6.177	112	12		√	
RC107	340.96	2566	6.521	112	13		√	
RC108	416.50	2594	9.204	134	14		√	
RC201	221.39	2408	6.428	127	8		√	
RC202	418.60	2418	8.237	139	8		√	
RC203	472.32	2465	8.783	136	5		√	
RC204	624.22	2553	8.705	148	8		√	
RC205	364.55	2160	6.458	149	9		√	
RC206	293.30	2482	7.410	128	13		√	
RC207	280.69	2477	7.878	131	7		√	
RC208	385.72	2737	8.72	124	9		√	

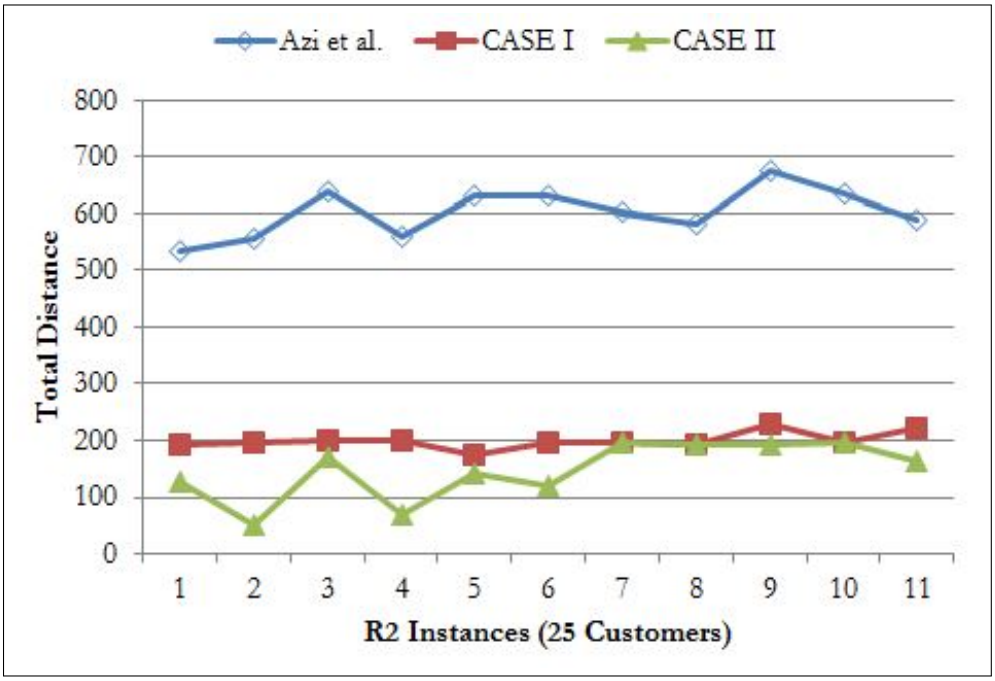

Figure 4.6: Result Comparison for total distance with Azi *et al.* (2007) from R2 instances

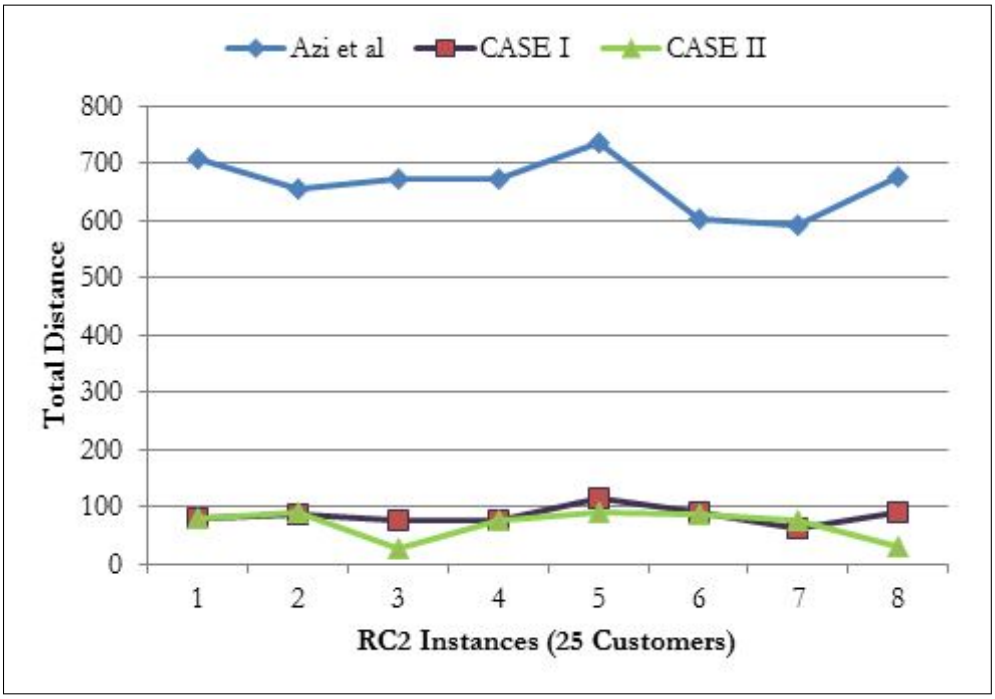

Figure 4.7: Result Comparison for total distance with Azi *et al.* (2007) from RC2 instances

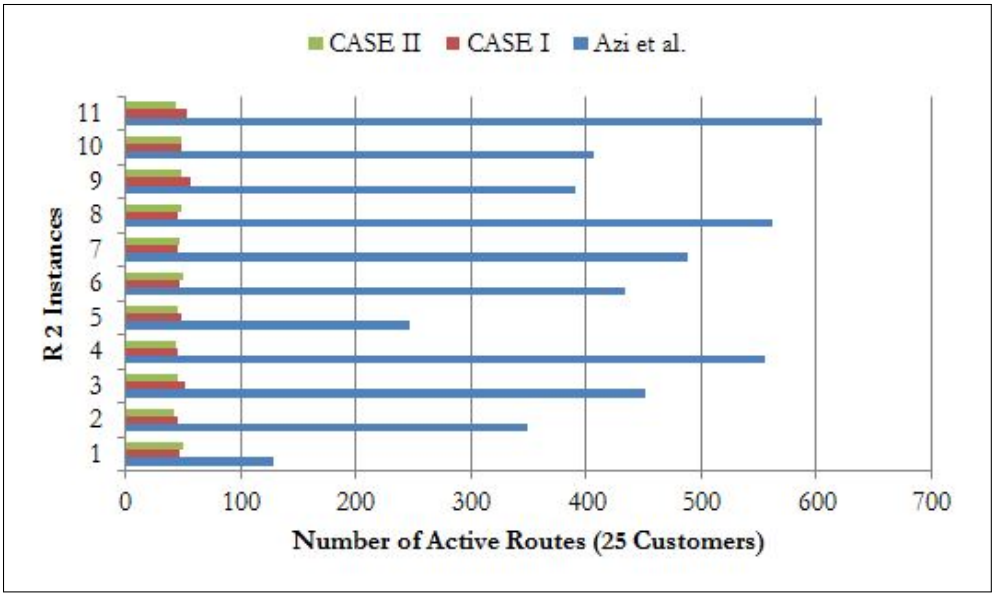

Figure 4.8: Result Comparison for number of active routes with Azi *et al.* (2007) from R2 instances

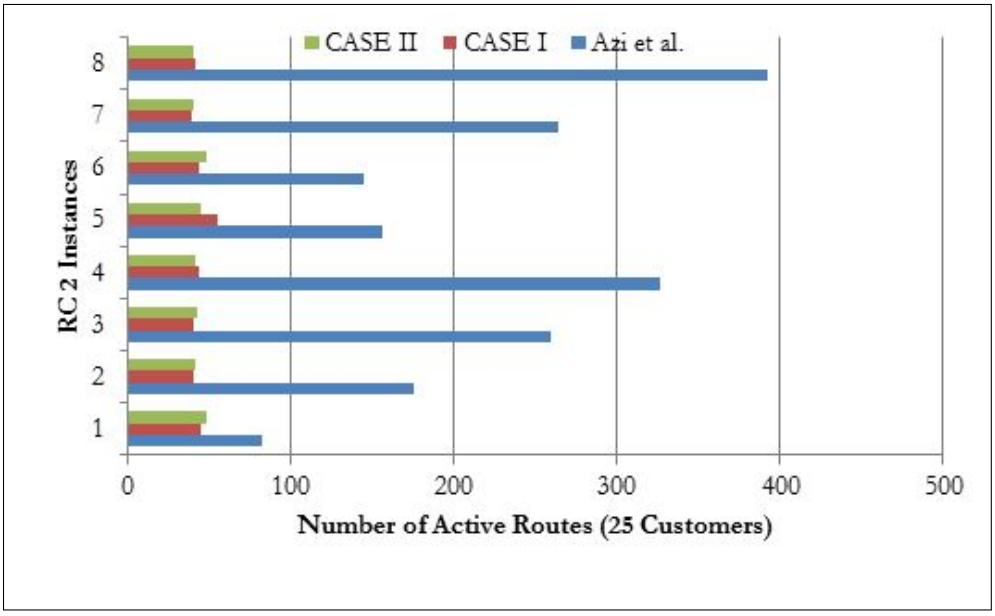

Figure 4.9: Result Comparison for number of active routes with Azi *et al.* (2007) from RC2 instances

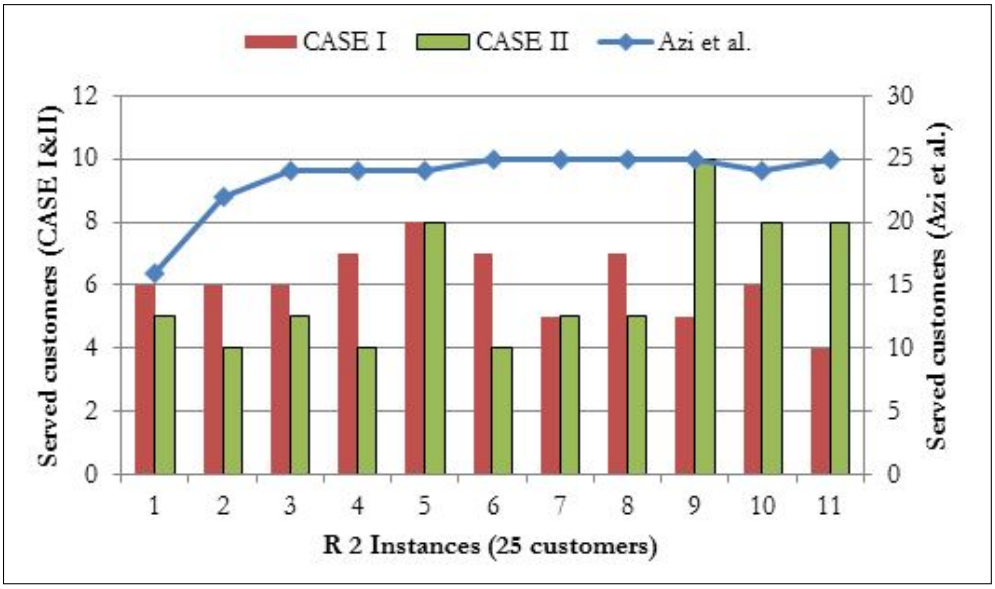

Figure 4.10: Result Comparison for number of served customers with Azi *et al.* (2007) from R2 instances

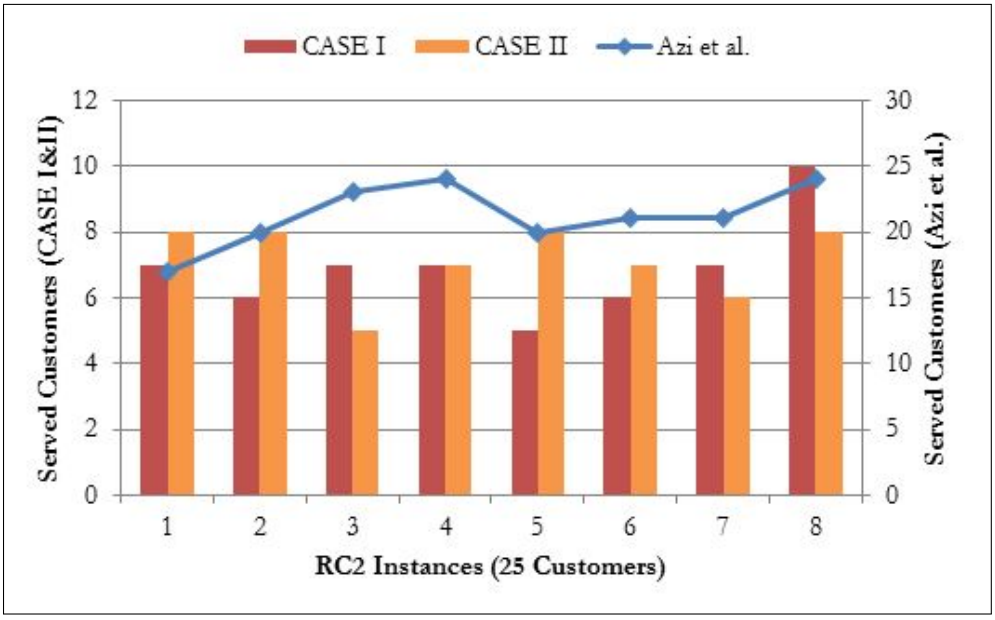

Figure 4.11: Result Comparison for number of served customers with Azi *et al.* (2007) from RC2 instances

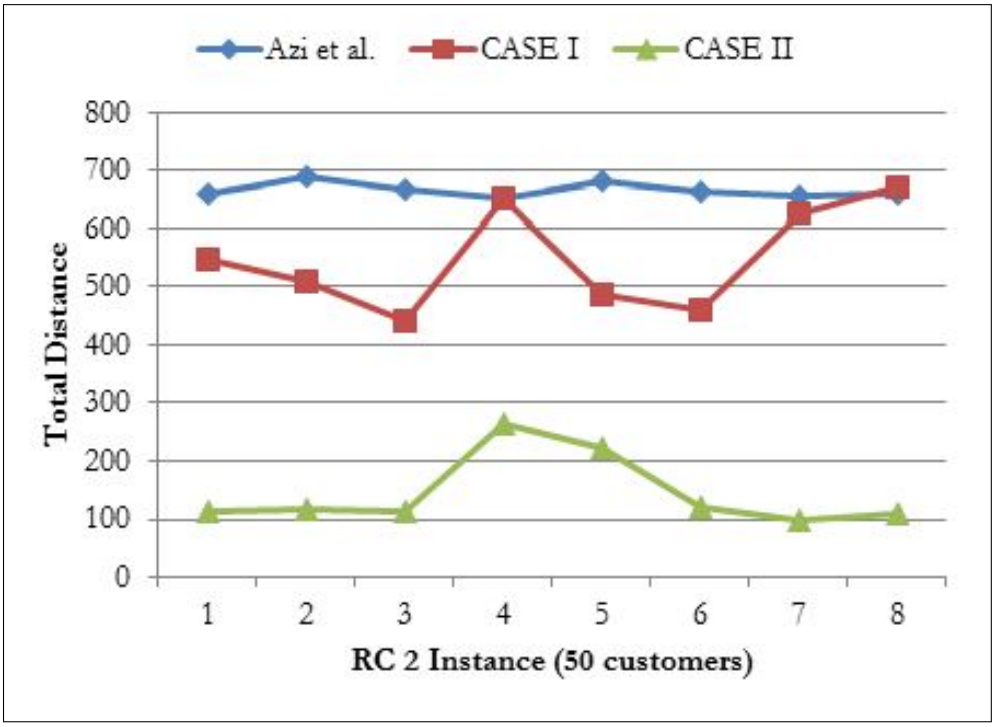

Figure 4.12: Result Comparison for total distance with Azi *et al.* (2007) from RC2 instances of 50 customers

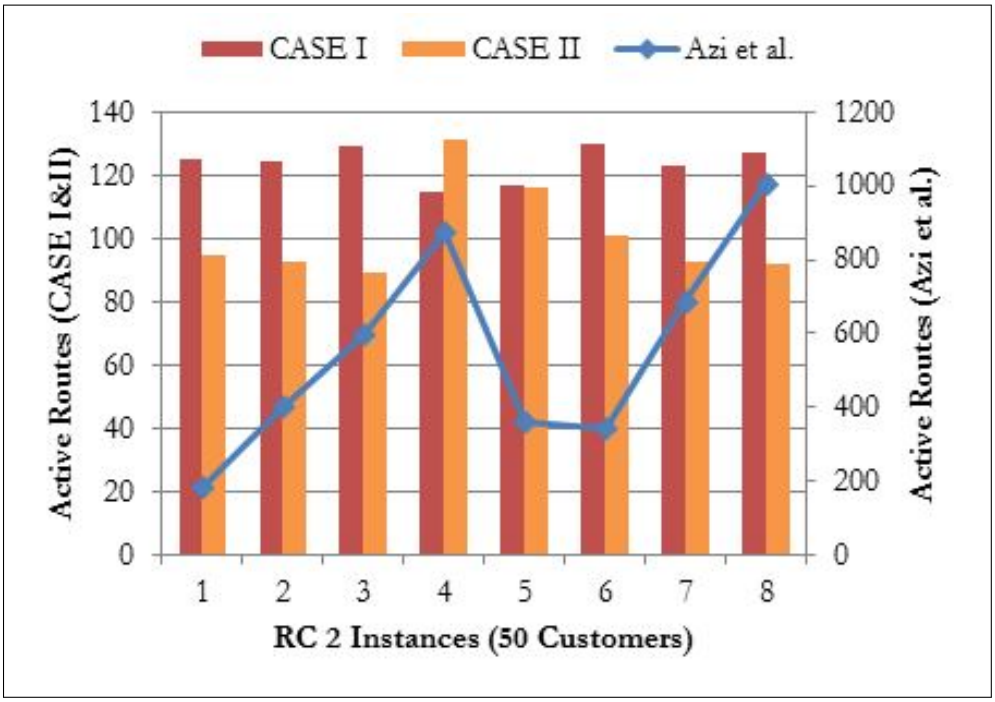

Figure 4.13: Result Comparison for active routes with Azi *et al.* (2007) from RC2 instances of 50 customers

Table 4.20: Result comparison with Azi *et al.*, (2007)

25 Customers at $t_{max} = 75$									
Test	Total Distance			Active Routes			Served Customers		
Instance	Azi et al	Case I	Case II	Azi et al	Case I	Case II	Azi et al	Case I	Case II
R201	535.77	192.22	126.99	129	47	50	16	6	5
R202	556.41	197.14	51.42	349	46	42	22	6	4
R203	641.13	199.37	170.90	452	51	46	24	6	5
R204	557.77	201.61	70.43	556	45	43	24	7	4
R205	631.73	174.62	142.49	246	48	45	24	8	8
R206	633.42	195.79	118.53	434	47	50	25	7	4
R207	602.09	195.58	195.58	489	45	47	25	5	5
R208	582.61	194.25	194.16	561	45	49	25	7	5
R209	675.59	228.44	194.21	391	56	48	25	5	10
R210	635.48	196.06	195.95	406	48	49	24	6	8
R211	588.01	221.60	163.04	605	54	43	25	4	8
RC201	707.01	80.59	80.59	83	45	48	17	7	8
RC202	654.75	89.32	91.95	175	40	42	20	6	8
RC203	673.20	77.43	26.87	260	40	43	23	7	5
RC204	673.93	77.46	76.22	327	44	42	24	7	7
RC205	734.59	117.47	90.84	156	55	45	20	5	8
RC206	601.81	89.88	89.44	145	44	48	21	6	7
RC207	593.99	62.94	76.26	264	39	40	21	7	6
RC208	678.09	91.46	33.24	392	42	41	24	10	8
50 Customers at $t_{max} = 75$									
R201	599.12	503.43	307.42	560	130	106	27	9	7
R202	651.09	582.74	339.41	2233	121	104	32	9	6
R203	NA	739.39	340.52	3248	130	112	NA	7	9
R204	NA	758.04	369.49	4535	124	121	NA	6	9
R205	645.56	515.41	305.24	1434	131	109	31	12	9
R206	629.29	779.21	371.28	3069	130	114	34	7	6
R207	NA	714.90	402.89	3682	118	115	NA	7	5
R208	NA	669.02	380.67	4548	127	115	NA	6	7
R209	639.24	577.95	283.61	2531	129	110	32	12	5
R210	NA	610.96	310.01	3160	139	103	NA	5	6
R211	NA	507.92	319.85	5067	125	122	NA	8	6
RC201	660.68	547.01	112.80	182	125	95	21	9	11
RC202	689.55	510.74	116.20	403	124	93	25	9	12
RC203	668.27	440.55	114.93	595	129	89	27	8	11
RC204	653.95	651.97	264.02	873	115	131	29	6	9
RC205	681.89	484.65	221.96	361	117	116	24	8	3
RC206	664.86	460.16	119.51	342	130	101	25	7	12
RC207	655.43	624.47	99.67	687	123	93	26	9	12
RC208	658.28	669.42	108.64	1003	127	92	28	7	12

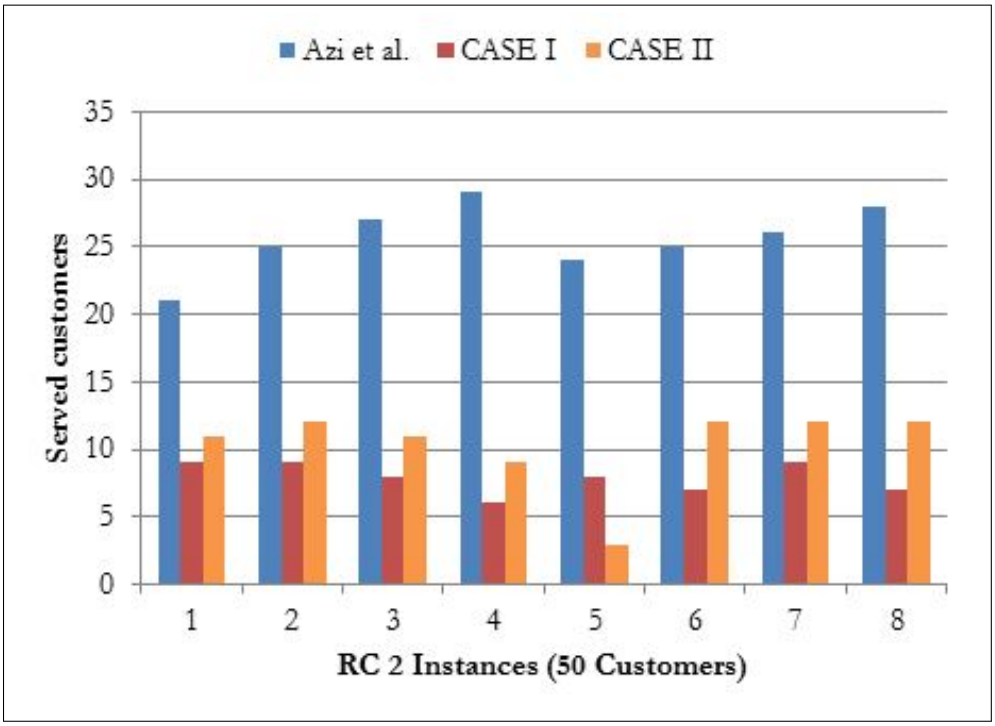

Figure 4.14: Result Comparison for number of served customers with Azi *et al.* (2007) from RC2 instances of 50 customers

CHAPTER FIVE

DISCUSSION

5.1 Introduction

In this chapter, the results obtained in chapter four are discussed. Since two models were proposed and implemented, the discussions are carried out under two separate sections as follows.

5.2 Discussion of Results on the Implementation of SWC Model

In computing the results in Tables 4.1-4.11, five different threshold distances were defined to capture different scenarios. These distances were measured in meter in conformity with the data obtained with the GDMC. They are $100/125/150/175/200m$. The three types of waste are denoted by $t1$, $t2$ and $t3$ respectively. In Tables 4.1-4.5, the full solution for all the problem instances are reported. In each instance, the %gap between the current candidate collection sites and the objective values representing the optimal number of collection sites are also reported. The execution time is the sum of the input, solve and output time (all in seconds). The last columns in these tables contain the information about the number of cuts the solver applied on the problem. Applied cuts in each problem instance are displayed in Tables 4.9-4.11. These cuts are necessary to prevent non-integer solutions which would have resulted for a case of continuous relaxation of the problem. Cuts are simple constraints that are generated and embedded into the problem by the solver in such a way that they are kept valid for all instances of sub-problems. In all, six different cuts were reported.

Three other tables, Tables 4.6-4.8, refer to the results on optimal allocation of containers to the activated sites containing a comparison of the current and new allocations. For all the computational experiments, three different capacities for the different containers (corresponding to the different waste types) are considered: $50/60/70kg$. In computing the results, no time limit was imposed on the solver. This was because the focus was not on the efficiency of the applied method but rather on the objective of the model. The various results are presented in Tables 4.1-4.11.

Summarily, the results showed that there was an overall reduction in the number of collection sites to be opened. As the threshold distance increases, the number of open facilities reduces which was quite as expected. This result was illustrated in Figure 4.1. The result obtained for container allocation also revealed drastic fall in the number of containers allocated to the various activated sites. This was illustrated in Figure 4.2. CA denotes current allocations

while NA/? represents new allocations for different capacities. Generally, there is more than 60% reduction in the allocation.

5.3 Discussion of Results on the Implementation of SWD Model

To each arc in the SWD system, there was an associated accessibility ratio, R_{fk}. The values of these ratios were randomly generated between 0 and 1. As indicated in the model, we assumed that every arc originating from the depot and ending at the depot have R_{fk} that satisfied $\theta_{fk} = 1$ to ensure that a vehicle leaves and returns to the depot. Furthermore, to show flexibility of the model, tests were conducted with two different bounds on R_{fk} (i.e., $R_{fk} \geq 0.4$ and $R_{fk} \geq 0.5$). The value of M in each of the test is 1000.

The implementation of the SO algorithm was performed on the AMPL optimization system. However, the IP solver, CPLEX, used in implementing the SWC model could not handle the SWD model because of the nonlinearity in the objective as well as some of the constraints. This shortcoming prompted the use of another solver called MINOS which is more suitable for nonlinear optimization problems. The version of the solver used is MINOS 5.51.

Two customer sizes of 25 and 50 were implemented in the computations. The **C1** and **C2** instances were discarded. Only cases of **R1**, **R2**, **RC1** and **RC2** were considered because of the combined features of clusterization and randomization of the instances. In general, three solution types were reported by the solver. This solution report is denoted by **SSR** in the various tables presented for results. The first report is on optimality of the solution denoted by **Opt**, meaning optimal solution was found. The second denoted **ExIter** means excessive iteration, that is, there are too many major iterations leading to solution, and thirdly, a case where current solution cannot be improved denoted **CPNI**. Other result identifiers in the tables are: **NLO**- nonlinear objective, **TD**- total distance, **Iter**- number of iterations, **Exec Time**- solver execution time, **AA**- number of active arcs and **NSC**- number of served customers.

The results presented in Tables 4.12-4.19 are computed based on R1, R2, RC1 and RC2 instances; 25 and 50 customers; and two different values of R_{ij}: $R_{ij} \geq 0.4$ and $R_{ij} \geq 0.5$. In Table 4.12, the results on test with the nonlinear case (CASE I) produced 28 optimal solutions out of the total 39 problems solved. These results further show that an average of 4, 3, 6 and 5 customers were visited on each trip for the instances respectively. For the case of $R_{ij} \geq 0.5$ (Table 4.13), 26 problems were solved to optimality. However, in this case, more customers were visited: 6, 6, 8 and 7 respectively.

In this research, the algorithm was not restricted to a limited number of iterations, hence the problem instances with 50 customers produces more iterations forcing the solver to generate

reports indicating either too many iterations or that current point could not be improved. Tables 4.16-4.19 represent results with the linear objective (CASE II) and all the problems with 25 customers were optimally solved when $R_{ij} \geq 0.5$, with two less when $R_{ij} \geq 0.4$. Optimal solutions were obtained for few problems with 50 customers. It is easily observed that when $R_{ij} \geq 0.5$ more customers are served compared to when $R_{ij} \geq 0.4$. This shows the effect of the accessibility ratio on route construction. In fact, the distances vary for the two values of R_{ij} and the two cases of the formulation.

Closely related to this work is the study conducted by Azi *et al.* (2007) where a VRPTW was considered such that a single vehicle was allowed to make multiple trips. The computational test in the study was also based on the Solomon's instances. A comparison was made with the results found in the work. The routes were constructed in two phases and so the sums were found to represent the total number of active routes. The algorithm proposed in Azi *et al.* (2007) was not applied to the R1 and RC1, hence, the comparison was restricted to the R2 and RC2. In Table 4.20 this comparison is presented for the total minimum distance, active routes and number of served customers. Only the $R_{ij} \geq 0.5$ cases of the two problem cases was considered. Different values of t_{max} (maximum travel time) were used in Azi *et al.*, (2007). We only compared with the results based on the value of $t_{max} = 75$. Figures 4.6-14 showed the comparisons very clearly.

In Figures 4.6-4.7, it is obvious that the difference between the maximum total distance of the problem proposed here and the minimum total distance of Azi *et al.* is 300 and 500 for the R2 and RC2 instances respectively. The same outcome is observed in Figures 4.8-4.9 where fewer number of arcs were active in our model as expected. However, in Figure 4.10, the Azi *et al.* (2007) results showed that more customers were visited. Normally, one would expect this since their model does not consider the attribute of each arc prior to route planning process. The same can be said of Figure 4.14 with 50 customers.

CHAPTER SIX

CONCLUSIONS AND RECOMMENDATIONS

6.1 Summary

The research reported in this thesis was carried out with the aim of developing effective solid waste collection and disposal systems in urban residential areas. The motivation for embarking on the work arose from gaps identified in literature (especially from the work of Ghiani *et al.* 2012) and the impacts of faulty roads on waste vehicles in most developing low-income countries. Finding waste generation data, either from local authorities or in literature, is much easier especially when it relates to regional or national data. However, waste generation data for individual building or resident rarely appear in scientific papers. Therefore, many researches on waste collection have always revolve round regional data. Considering the fact that wastes are generated on per-capita basis, designing a collection system that captures this form of data therefore becomes so imperative. Furthermore, owing to the region where this research emanates from, recurrent infrastructural concerns coupled with high costs of maintaining waste collection vehicles has made it so important and urgent to introduce new parameters into the disposal phase of waste management. The new parameter describes the attributes of the roads in a collection area.

Based on these two concerns, that is, per-capita waste generation and infrastructures, two models were developed independently to handle the problem of solid waste collection and disposal (SWCD). The first model was designed such that a customer (a house, individual resident or any other source where waste is generated) is assigned to the closest site from the set of minimized collection sites. The model was formulated as facility location problems (FLP). The second model, where the road attribute variable was introduced seeks to find a set of optimal routes for a collection vehicle such that the total distance covered on each operational trip is minimized. This model was formulated as a vehicle routing problem (VRP).

6.2 Conclusions

This work was carried out to address the challenges of waste collection and disposal in urban residential areas. Specifically, models were developed for effective location of waste collection sites and the routing of waste collection vehicles. Experience and observation have shown that waste collection from the initial sources of generation in most regions especially in low-income developing nations is based on ad-hoc location of waste containers or dump-sites.

Also, low budgetary allocation for road infrastructures resulting in many faulty roads has a huge effect on disposal vehicles. Hence, owing to these two shortcomings, two models have been proposed and solved to effectively handle these backdrops. The objectives of the proposed models are to: (i) minimize the total number of active collection sites in a given residential area (ii) find the optimal assignment of customers to these active sites (iii) find the optimal allocation of containers to the active site (iv) find the minimum active route for a waste collection vehicle such that the total travel distance is minimized.

Objectives (i)-(iii) are captured in the model for site location (SWC model) while (iv) is mainly addressed in the model for vehicle routing (SWD model).

6.2.1 Empirical Findings

Briefly, the various findings in this study are reported in this section. Each subsection is dedicated to each of the objectives stated above.

6.2.1.1 Findings on optimal locations of collection sites

Results obtained showed that the model effectiveness increases as the number of potential collection sites used increases. For instance, comparing between the case of 100 and 500 candidate sites (when the threshold distance is 100), 80 and 95 sites are to be activated respectively. However, considering cost effectiveness, it can be concluded that the model is more suitable for large-scale problem instances. If real life data are available, the model may be implemented for a city or a region rather than for a small residential area with few houses. Furthermore, the average number of activated sites for the threshold distance 150m was 49.2. This prompted the consideration of Solomon's 25 and 50 customers instances in the implementation of the vehicle routing model.

6.2.1.2 Findings on optimal assignment of clusters to collection sites

A brief sample of the results on assignment of clusters to the collection sites can be found in Appendix C. If the assignments are uniformly computed, it will be observed that problem instances with 100 and 200 potential sites (and threshold distance of 100 and 125 meters) produce maximum of 3 cluster assigned to a site which is not cost effective. The trend is that, as the problem size increases, the number of assignment also increases. For instance, for medium-sized problems (400 and 500 clusters), there is a uniform assignment of 8 clusters per activated site.

6.2.1.3 Findings on optimal allocation containers to collection sites

Starting from the first problem instance, the number of containers for each type of waste were doubled. Three different sizes of containers are considered and the results showed that a slight difference occurs in the number of new allocations of each container size. This may be so because of the random choice of the number available for each capacity. Also, there was a drastic reduction in the number of containers allocated to the activated sites.

6.2.1.4 Findings on route construction and minimal travel distance

The model proposed to minimize travel distance during the disposal phase of collection forms a major contribution of this study. A new parameter addressing road accessibility was introduced with the aim of preventing waste collection vehicles from using faulty roads that may cause frequent breakdowns thereby shortening their life span. This model was studied in two forms: linear and nonlinear objective functions. The numerical results obtained for the decision variables in the two cases were compared with a previous work by Azi *et al.* (2007) using the test instances of Solomon (1987) with 25 and 50 customers. In this phase of the study, customers represent the activated collection sites. The best results were obtained with Case II (linear) for 25 customers and $R_{fk} \geq 0.5$. In this case all the problems were solved to optimality (see Table 4.12). The RC instances produced minimal distances and more number of served customers. Generally, the results obtained produced minimal travel distances compared to the Azi *et al.* (2007) case. Fewer number of sites were visited with model II considering the inclusion of the variable on road attribute. This shortfall may be addressed by introducing tighter restrictions on the model and careful evaluation of the various factors contributing to the accessibility of a road.

6.3 Research Contributions

The following are the contributions of this research work

i. Two new models were formulated to address the problems associated with solid waste collection and disposal. The models were essentially designed and implemented to assist policy makers in low-income developing nations on the best approach to low cost waste collection and disposal.

ii. Studies on finding optimal locations for collection sites through clustering have only used waste generation data that are fundamentally meant for a whole region (city, state or country). However, in this study, the collection model have been so formulated to

handle the quantity of waste generated from each cluster and for different wastes. Thus, previous studies on the locations of waste collection facilities have been improved.

iii. Two new parameters measuring road accessibility were introduced in the disposal model. Road accessibility is a major challenge in many low-income developing countries and because the targets of this research are urban areas in these regions, the motivation for the inclusion of this parameter became necessary. This study has shown that a model with this parameters will result in decrease in the total distance traveled by waste collection vehicles with a considerable number of sites visited.

iv. Two comprehensive literature reviews were conducted on the two classes of problem that were used to formulate the models proposed. These surveys can serve as guides for interested researchers on the current trends in this aspect of combinatorial optimization.

6.4 Recommendations

Based on the findings reported in this thesis, the following key decisions are recommended for waste management policy makers and researchers in SWM.

i. The challenges facing local authorities saddled with the task of SWCD are more often related to ad-hoc placement of containers at wrong locations either too far from the sources of waste generation or not easily accessible by collection vehicles. The models proposed in this study are therefore recommended for adoption by policy makers for finding cost-effective locations for solid waste collection sites.

ii. Regions which are badly characterized by inaccessible roads should adopt simple approaches such as Push Carts, Wheel Barrows, etc, to collect wastes on certain routes where accessibility constraints are violated. Available vehicles should be used only for routes obtained from the implementation of the disposal model.

iii. Governments should endeavor to direct more efforts toward making waste generation data available, especially quantity generation from different locations. This will strengthen research on the assignment and allocation of clusters and containers to the collection sites respectively.

6.5 Possible Areas of Future Research

Observation from the reviewed materials, computational experiments conducted and the research limitations mentioned in Chapter One have raised some possible research directions

that may be exploited in the future. Some of these identified areas are listed below.

i. The need to test the models on larger problem instances from literature. So far, the test conducted on Model I are based entirely on randomly generated data and for Model II, instances used are only for 25 and 50 customers sizes. The computations are limited to these medium-sized instances because of inadequate computing resources in terms of machine and softwares. Hence, effort will be made at implementing the models with large-scale problems in order to compare their efficiency with other related models tested on similar instances of problem.

ii. In this study, a clustering approach similar to the capacitated clustering has been used. However, there are several other clustering techniques that have been used, although in other context, by authors, some of which can be adopted in formulating the problem for the location of the collection sites. Future work may involve using one or more of these techniques to design more effective collection system.

iii. The LR method only produced the lower bound values on the optimal values for the set of decision variables. Several heuristic and metaheuristic procedures are now available that can interact with the lower bound solutions produced from the Lagrangian problem within the subgradient optimization scheme. It will be desirable to construct this near-optimal solution seeking methods to reinforce the effectiveness of the models.

iv. The data values for the parameter on accessibility ratio were randomly generated without any special consideration of various contributory factors. Careful consideration of these factors will guide the manner in which this parameter is scaled to better represent a real-life system.

v. Solution to Model II consists of relaxing some kickback parameters to reduce the complexity of the problem due to limited computing environment. These parameters will be re-introduced in the future to increase the robustness of the model and to better represent governmental policies on daily operations.

REFERENCES

Abyasi-Sani, R. and Ghanbari, R. (2016). An efficient tabu search for solving the uncapacitated single allocation hub location problem. *Computers and Industrial Engineering* **93:** 99-109.

Adewumi, A. O. (2015). Lecture note on optimization and modeling. University of Kwazulu-Natal, Westville Campus, Durban, South Africa.

Adewumi, A. O. and Arasomwan, M. A. (2016). On the performance of particle swarm optimization with(out) some control parameters for global optimization. *Int. J. Bio-Inspired Computation* **8(1):** 14-32.

Ai, T.J. and Kachitvichyamikul, V. (2009). Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem. *Computers and Industrial Engineering* **56:** 380-387.

Albareda-Sambola, M., Fernandez, E. and Laporte, G. (2014). The dynamic multi-period vehicle routing problem with probabilistic information. *Computers and Operations Research* **48:** 31-39.

Alegre, J., Laguna, M. and Pacheco, J. (2007). Optimizing the periodic pickup of raw materials for a manufacturer of auto parts. *European Journal of Operational Research* **179:** 736-746.

Anderson, G., Francis, L. R. Normark, T. and Rayco, M. B. (1998). Aggregation method experimentation for large scale network location problems. *Location Sci.* **6:** 25-39.

Angelelli, E. and Speranza, M. G. (2002). The periodic vehicle routing problem with intermediate facilities. *European Journal of Operational Research* **37(2):** 233-247.

Angelelli, E., Bianchessi, N., Mansini, R. and Speranza, M. G. (2009). Short term strategies for a dynamic multi-period routing problem. *Transportation Research Part C: Emerging Technologies* **17(2):** 106-119.

Arasomwan, M. A. and Adewumi, A. O. (2014). Improved Particle Swarm Optimization

with a Collective Local Unimodal Search for Continuous Optimization Problems. *The Scientific World Journal* **2014:** 1-23.

Archetti, C., Savelsbergh, M. W., and Speranza, M. G. (2006). Worst-case analysis for split delivery vehicle routing problems. *Transportation science* **40(2):** 226-234.

Archetti, C., Jabali, O. and Speranza, M. G. (2015). Multi-period vehicle routing problem with due date. *Computers and Operations Research* **61:** 122-134.

Armas, J. and Melian-Batista, B. (2015). Variable neighbourhood search for dynamic rich vehicle routing problem with time windows. *Computers and Industrial Engineering* **85:** 120-131.

Augerat, P., Belenguer, J.M., Benavent, E., Corberan, A., Naddef, D. and Rinaldi, G. (1995). Computational results with a branch-and-cut code for the capacitated vehicle routing problem. *Research report RR949-M, ARTEMIS-IMAG*, France.

Azi, N., Gendreau, M. and Potvin, J.-Y. (2007). An exact algorithm for a single-vehicle routing problem with time windows and multiple routes. *European Journal of Operational Research* **178(3):** 755-766.

Azi, N., Gendreau, M. and Potrin, J.-Y. (2010). An exact algorithm for a vehicle routing problem with time windows and a multiple use of vehicles. *European Journal of Operational Research* **202:** 756-763.

Badran, M. F. and El-Hagar, S. M. (2006). Optimization of municipal solid waste management in Port Said - Egypt. *Waste Management* **26:** 534-545.

Baldacci, R., Mingozzi, A. and Roberti, R. (2012). Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research* **218(1):** 1-6.

Balseiro, S. R., Loiseau, I. and Ramonet, J. (2011). An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows. *Computers and Operations Research* **38:** 954-966.

Banerjea-Brodeur, M., Cordeau, J. F., Laporte, G., and Lasry, A. (1998). Scheduling linen

deliveries in a large hospital. *Journal of the Operational Research Society* **49(8):** 777-780.

Beasley, J. E. (1993). Lagrangean relaxation. In *Modern heuristics techniques for combinatorial problems* (pp. 243-303). John Wiley & Sons. Inc.

Beasley, J. E. (1990). OR-Library: distributing test problems by electronic mail. *Journal of Operational Research Society* **41(11):** 1069-1072.

Belenguer, J. M., Martinez, M. C., and Mota, E. (2000). A lower bound for the split delivery vehicle routing problem. *Operations Research* **48(5):** 801-810.

Belfiore, P. P. and Yoshizaki, H. T. Y. (2006). Scatter search for vehicle routing problems with heterogeneous fleet, time windows and fractional deliveries. *Production [online]* **16:** 455-469.

Belfiore, P., Tsugunobu, H., and Yoshizaki, Y. (2008). Scatter search for vehicle routing problem with time windows and split deliveries. *Vehicle Routing Problem* 1-14.

Belfiore, P. and Yoshizaki, H. T. Y. (2013). Heuristic methods for the fleet size and mix vehicle routing problem with time windows and split deliveries. *Computers and Industrial Engineering* **64:** 589-601.

Beltrami, E. J., and Bodin, L. D. (1974). Networks and Vehicle Routing for Municipal Waste Collection. *Networks* **4:** 65-94.

Benjamin, A. M. and Beasley, J. E. (2010). Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities. *Computers Operations Research* **37:** 2270-2280.

Beresnev, V. (2013). Branch-and-bound algorithm for a competitive facility location problem. *Computers and Operations Research* **40:** 2062-2070.

Berman, O., Krass, D. and Drezner, Z. (2003). The gradual covering decay location problem on a network. *European Journal of Operational Research* **151(3):** 474-480.

Bertimas, D. and Tsitsiklis, J. (1993). Simulated annealing. *Statistical Science* **8(1):** 10-15.

Bertsekas, D. P., Nedic, A. and Ozdaglar, A. E. (2003). Convex analysis and optimization. Athena Scientific, Belmont, Massachusetts.

Bhattachariya, R. K. (2013). Introduction to genetic algorithms. INTERNET: http:/www.iitg.ernet.in/rkbc/CE602/CE602/GeneticAlgorithms.pdf. Cited 08 Feb 2016.

Blakeley, F., Argello, B., Cao, B., Hall, W., and Knolmajer, J. (2003). Optimizing periodic maintenance operations for Schindler Elevator Corporation. *Interfaces* **33(1):** 67-79.

Bock, A. and Sanita, L. (2013.) On the approximability of two capacitated vehicle routing problems. *Electronic Notes in Discrete Mathematics* **41:** 519-526.

Bortfeldt, A. (2012). A hybrid algorithm for the capacitated vehicles routing problem with three-dimensional loading constraints. *Computers and Operations Research* **39(9):** 2248-2257.

Bouthillier, A. L. and Crainic, T. G. (2005). A cooperative parallel metaheuristic for the vehicle routing problem with time windows. *Computer and Operations Research* **32:** 1685-1708.

Branchini, R. M., Armentano, V. A. and Lokketangen, A. (2009). Adaptive granular local search heuristic for dynamic vehicle routing problem. *Computers and Operations Research* **36:** 2955-2968.

Buhrkal, K., Larsen, A. and Ropke, S. (2012). The waste collection vehicle routing problem with time windows in a city logistics context. *Procedia - Social and Behavioral Sciences* **39:** 241-254.

Burke, E. K. and Kendall, G. (2005). Introduction. In: Burke E. K. and Kendall G. (eds). Search methodology: introductory tutorials in optimization and decision support techniques. Springer Science + Business Media, LLC, ISBN: 0-0387-23460-8.

Calvete, H. I., Gale, C., Oliverros, M.J. and Sanchez-Valverde, B. (2007). A goal programming approach to vehicle routing problem with soft time windows. *European Journal of Operational Research* **177:** 1720-1733.

Chen, C.-H. and Ting, C.-J. (2008). Combining Lagrangian heuristic and Ant Colony Sys-

tem to solve the Single Source Capacitated Facility Location Problem. *Transportation Research Part E: Logistics and Transportation Review* **44(6):** 1099-1122.

Cheng, C. B. and Wang, K. P. (2009). Solving a vehicle routing problem with time windows by a decomposition technique and genetic algorithm. *Expert Systems with Applications* **36:** 7758-7763.

Chiang, T.-C. and Hsu, W.-H. (2014). A knowledge-based evolutionary algorithm for the multiobjective vehicle routing problem with time windows. *Computers and Operations Research* **45:** 25-37.

Choi, E. and Tcha, D. W. (2007). A column generation approach to the heterogeneous fleet vehicle routing problem. *Computers and Operations Research* **34(7):** 2080-2095.

Christiansen, C. H. and Lysgaard, J. (2007). A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research Letters* **35(6):** 773-781.

Christofides, N., Mingozzi, A., and Toth, P.(1979). The vehicle routing problem. In: Christofides, N., Mingozzi, A., Toth, P., and Sandi, C. (editors). *Combinatorial optimization.* Chichester: Wiley, 315-338.

Christofides, N. and Beasley, J. E. (1984). The periodic routing problem. *Networks* **14(2):** 237-256.

Clarke, G. and Wright, J. (1964). Scheduling of vehicle from central depot to number of delivery points. *Operations Research* **12(4):** 568-581.

Clausen, J. (1999). Branch and bound algorithms-principles and examples. Department of Computer Science, University of Copenhagen, 1-30.

Coffey, M. and Coad, A. (2010). Collection of Municipal Solid Waste in Developing Countries. United Nations Human Settlements Programme (UN-HABITAT), Kenya.

Colorni, A., Dorigo, M. and Maniezzo, V.(1991). Distributed Optimization by ant colonies. *First European Conference on Artificial Life* **1:** 134-142.

Contreras, I., Diaz, J. A. and Fernández, E. (2009). Lagrangian relaxation for the capacitated hub location problem with single assignment. *OR Spectrum* **31(3):** 483-505.

Cordeau, J. F., Gendreau, M. and Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* **30(2):** 105-119.

Cordeau, J. F., Laporte, G. and Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational research society* **52(8):** 928-936.

Dantzig, R. and Ramzer, J. (1959). The truck dispatching problem. *Management Science* **6(1):** 80-91.

Daskin, M. S. (1995). Network and discrete location: models, algorithms and applications. Wiley, New York.

Dayarian, I., Crainic, T. G., Gendreau, M. and Rei, W. (2015). A branch-and-price approach for a multi-period vehicle routing problem. *Computers and Operations Research* **55:** 167-184.

Dhahri, A., Zidi, K. and Ghedira, K. (2014). Variable neighborhood search based set covering ILP model for the vehicle routing problem with time windows. *Procedia Computer Science* **29:** 844-854.

Dorigo, M. and Stützle T. (2004). Ant colony optimization. MIT Press, USA.

Drezner, Z. and Wesolowsky, G. O. (2001). On the collection depots location problem. *European Journal of Operational Research* **130:** 510-518.

Dror, M., and Trudeau, P. (1989). Savings by split delivery routing. *Transportation Science* **23(2):** 141-145.

Dror, M. and Trudeau, P. (1990). Split vehicle routing. *Naval Research Logistics* **37(3):** 383-402.

Dror, M., Laporte, G., and Trudeau, P. (1994). Vehicle routing with split deliveries. *Discrete Applied Mathematics* **50(3):** 239-254.

Eberhart, R. C. and Kennedy, J. (1995). A new optimizer using particle swarm theory. *Proceedings of the sixth international symposium on micro machine and human science* **1:** 39-43.

Erkut, E., Karagiannidis, A., Perkoulidis, G. and Tjandra, S. A. (2008). A multicriteria facility location model for municipal solid waste management in North Greece. *European Journal of Operational Research* **187:** 1402-1421.

Euchi, J., Yassine A. and Chabchoub, H. (2015). The dynamic vehicle routing problem: solution with hybrid metaheuristic approach. *Swarm and Evolutionary Computation* **21:** 41-53,

Fabri, A. and Recht, P. (2006). On dynamic pickup and delivery vehicle routing with several time windows and waiting times. *Transportation Research Part B* **40:** 335-350.

Farahani, R. Z. and Hekmatfar, M. (eds) (2009). Facility location: concepts, models, algorithms and case studies. Contributions to Management Science, Physica-Verlag, Heidelberg.

Ferrucci, F. and Bock, S. (2015). A general approach for controlling vehicle en-route diversions in dynamic vehicle routing problems. Transportation Research Part B **77:** 76-87.

Ferrucci, F., Bock, S. and Gendreau, M. (2013). A pro-active real-time control approach for dynamic vehicle routing problems dealing with the delivery of urgent goods. *European Journal of Operational Research* **225:** 130-141.

Fisher, M. L. (1994). Optimal solution of vehicle routing problems using minimum k-trees. *Operations Research* **42:** 626-642.

Fisher, M. L. (2004). The lagrangian relaxation method for solving integer programming problems. *Management Science* **50(12):** 1861-1871.

Fisher, M. L., and Jaikumar, R. (1981). A generalized assignment heuristic for vehicle routing. *Networks* **11(2):** 109-124.

Fouard, C. and Malandain, G. (2005). 3-D Chamfer distance and norms in anisotropic grids. *Image Vision Compt* **23:** 143-158.

Francis, P. and Smilowitz, K. (2006). Modeling techniques for periodic vehicle routing problems. *Transportation Research Part B* **40:** 872-884.

Francis, P., Smilowitz, K. R. and Tzur M. (2008). The periodic vehicle routing problem and its extensions. In: Golden, B., Raghavan, S., and Wasil, E. (editors). The vehicle routing problem: latest advances and new challenges. Springer, 73-102.

Freund, R. (1994). Professor George Dantzig: linear programming founder turns 80. SIAM News. Retrieved from http://www.stanford.edu/group/SOL/dantzig.html.

Frizzell, P. W. and Giffin, J. W. (1992). The Bounded Split Delivery Vehicle Routing Problem with Grid Network Distances. *Asia Pacific Journal of Operational Research* **9:** 101-116.

Frizzell, P. W., and Giffin, J. W. (1995). The split delivery vehicle scheduling problem with time windows and grid network distances. *Computers and Operations Research* **22(6):** 655-667.

Gehring, H. and Homberger, J. (1999). A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows. In: Mettinen, K., Makela, M. M., and Toivanen, J. (Eds.), *Proceedings of EUROGEN99-Short Course on Evolutionary Algorithms in Engineering and Computer Science* 57-64. Reports of the Department of Mathematical Information Technology, No. A 2/1999, University of Jyvaskyla, Finland.

Gendreau, M. and Potvin, J. -Y. (2005). Tabu search. In: Burke, E. K. and Kendall, G. (Eds.). Search methodology: introductory tutorials in optimization and decision support techniques. Springer Science + Business Media, LLC, ISBN: 0-0387-23460-8.

Gendreau, M., Ilori, M., Laporte, G. and Martello, S. (2006). A tabu search algorithm for a routing and container loading problem. *Transportation Science* **40:** 342-350.

Geoffrion, A. M. (1974). Lagrangian relaxation for integer programming. *Mathematical Programming Study* **2:** 84-114.

Ghannadpour, S. F., Noori, S., Tavakkoli-Moghaddam, R. and Ghoseiri, K. (2014). A multi-objective dynamic vehicle routing problem with fuzzy time windows: Model, solution and

application. *Applied Soft Computing* **14:** Part C, 504-527.

Ghiani, G., Laganá, D., Manni, E. and Triki, C. (2012). Capacitated location of collection sites in an urban waste management system. *Waste Management* **32(7):** 1291-1296.

Ghoseiri, K. and Ghannadpour, S. F. (2010). Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. *Applied Soft Computing* **10:** 1096-1107.

Gillet, B. and Miller, L. (1974). Heuristic algorithm for vehicle dispatch problem. *Operations Research* **22(2):** 340-349.

Glover, F. (1989). Tabu search-part I. *ORSA Journal on computing* **1(3):** 190-206.

Glover, F. (1997, October). A template for scatter search and path relinking. In *European Conference on Artificial Evolution* (pp. 1-51). Springer Berlin Heidelberg.

Godinho, M.T., Gouveia, L. and Magnanti, T.L.(2008). Combined route capacity and route length models for unit demand vehicle routing problem. *Discrete Optimization* **5:** 350-372.

Goel, A. and Gruhn, V. (2008). A general vehicle routing problem. *European Journal of Operational Research* **191(3):** 650-660.

Golden, B. L. (1975). Vehicle routing problems: formulations and heuristic solution techniques. Operations Research Center, MIT, Technical Report, No. 113.

Golden, B., Assad, A., Levy, L. and Gheysens, F. G. (1984). The fleet size and mix vehicle routing problem. *Computers and Operations Research* **11:** 49-66.

Golden, B. L., Wasil, E. A., Kelly, J. P. and Chao, I. M. (1998). Metaheuristics in vehicle routing. In: Craini T.G., Laporte G., editors. Fleet management and logistics, Boston: Kluwer, 33-56.

Google (2016). Google map of Eti-Osa, Lagos, Nigeria. Retrieved from https://www.google.com.ng/place/Eti-Osa.

Guignard, M. (2003). Lagrangean relaxation. *Top* **11(2):** 151-200.

Guner, A. R., Murat, A. and Chinnam, R. B. (2012). Dynamic routing under recurrent and non-recurrent congestion using real-time information. *Computers and Operations Research* **39(2):** 358-373.

Guta, B. (2003). Subgradient optimization methods in integer programming with an application to a radiation therapy problem. Ph.D Thesis, Dept. of Mathematics, University of Kasierslautern, Germany.

Gutierrez-Jarpa, G., Desaulniers, G., Laporte, G. and Marianov, V. (2010). A branch-and-price algorithm for the vehicle routing problem with deliveries, selective pickups and time windows. *European Journal of Operational Research* **206:** 341-349.

Hadjiconstantinou, E. and Baldacci, R. (1998). A multi-depot period vehicle routing problem arising in the utilities sector. *Journal of the Operational Research Society* 1239-1248.

Haghani, A. and Jung, S. (2005). A dynamic vehicle routing problem with time-dependent travel times. *Computers and Operations Research* **32:** 2959-2986.

Haimovich, M., Rinnooy Kan, A. H. G. and Stougie, L. (1988). Analysis of heuristics for vehicle routing problems. In: B. L. Golden, B. L., and Assad, A. A. (Eds.), Vehicle Routing: Methods and studies, North Holland, Amsterdam, 47-61.

Hamzadayi, A., Topaloglu, S. and Kose, S. Y. (2013). Nested simulated annealing approach to periodic routing problem of a retail distribution system. *Computers and Operations Research* **40:** 2893-2905.

Hashimoto, H., Yagiura, M. and Ibaraki, T. (2008). An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. *Discrete Optimization* **5:** 434-456.

Held, M., Wolfe, P. and Crowder, H. P. (1974). Validation of subgradient optimization. *Mathematical Programming* **6:** 62-88.

Ho, S. C., and Haugland, D. (2004). A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. *Computers and Operations Research* **31(12):** 1947-1964.

Homberger, J. and Gehring, H. (1999). Two evolutionary metaheuristics for the vehicle routing problem with time windows. In: G. Laporte and F. Semet (Eds.), Metaheuristics for location and routing problems. *Information Systems and Operational Research* **37:** (Special issue), 297-318.

Homberger, J. and Gehring, H. (2005). A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *European Journal of Operational Research* **162:** 220-238.

Hong, L. (2012). An improved LNS algorithm for real-time vehicle routing problem with time windows. *Computers and Operations Research* **39:** 151-163.

Hu, X, Sun, L. and Liu, L. (2013). A PAM approach to handling disruptions in real-time vehicle routing problems. *Decision Support Systems* **54:** 1380-93.

Hvattum, L. M., Lokketangen, A. and Laporte, G. (2007). A branch-and-regret heuristic for stochastic and dynamic vehicle routing problems. *Networks* **49(3):** 330-340.

Jans, R. and Degraeve, Z. (2008). A note on a symmetric set covering problem: the lottery problem. *European Journal of Operational Research* **186(1)** 104-110.

Jiang, J., Ng, K. M., Poh, K. L. and Teo, K. M. (2014). Vehicle routing problem with heterogeneous fleet and time windows. *Expert Systems with Application* **41:** 3748-3760.

Jin, J., Crainic, T. G. and Lokketangen, A. (2012). A parallel multi-neighborhood cooperative tabu search for capacitated vehicles routing problems. *European Journal of Operational Research* **222:** 441-451.

Jin, J., Crainic, T. G. and Lokketangen, A. (2014). A cooperative parallel metaheuristic for the capacitated vehicle routing problem. *Computers and Operations Research* **44:** 33-41.

Juan, A. A., Faulin, J., Ruiz, R., Barrios, B. and Caballe, S. (2010). The Sr-GCWS hybrid algorithm for solving the capacitated vehicle routing problem. *Applied soft computing* **10:** 215-224.

Kallehauge, B, Larsen, J. and Madsen, O. B. G. (2006). Lagrangian duality applied to the vehicle routing problem with time windows. *Computers and Operations Research* **33:**

1464-1487.

Kameyama, K. (2009). Particle swarm optimization-a survey. *IEICE transactions on information and systems* **92(7):** 1354-1361.

Ke, L. and Feng, Z.(2013). A two-phase in metaheuristic for the cumulative capacitated vehicle routing problem. *Computers and Operations Research* **40:** 633-638.

Kek, A. G. H., Cheu, R. L. and Meng, Q. (2008). Distance-constrained capacitated vehicle routing problems with flexible assignment of start and end depots. *Mathematical and Computer Modelling* **47:** 140-152.

Kim, B. I., Kim, S. and Sahoo, S. (2006). Waste collection vehicle routing problem with time windows. *Computers and Operations Research* **33:** 3624-3642.

Kim, D. G. and Kim, Y. -D. (2010). A branch and bound algorithm for determining locations of long-term care facilities. *European Journal of Operational Research* **206:** 168-177.

Kohl, N. and Madsen, O. B. G. (1997). An optimization algorithm for the vehicle routing problem with time windows based on Lagrangean relaxation. *Operations Research* **45:** 395-406.

Koskosidis, Y. A. and Powell, W. B. (1992). Clustering algorithms for consolidation of customer orders into vehicle shipments. *Transportation Research Part B: Methodological* **26(5):** 365-379.

Koskosidis, Y. A., Powell, W. B. and Solomon, M. M. (1992). An optimization-based heuristic for vehicle routing and scheduling with soft time window constraints. *Transportation Science* **26:** 69-85.

Krarup, J., and Pruzan, P. M. (1983). The simple plant location problem: survey and synthesis. *European Journal of Operational Research* **12:** 36-81.

Kuo, R.J., Zulvia, F. E. and Suryadi, K. (2012). Hybrid particle swarm optimization with generic algorithm for solving capacitated vehicle routing problem with fuzzy demand: A case study on garbage collection system. *Applied Mathematics and Computation* **219:** 2574-2588.

Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* **59(3):** 345-358.

Larsen, A. (2001). The dynamic vehicle routing problem, Ph.D Thesis, Technical University of Denmark (DTU).

Larsen, A., Madsen, O. B. G. and Solomon, M. M. (2008). Recent developments in dynamic vehicle routing systems. In: Golden, B., Raghavan, S., and Wasil, E. (editors). The vehicle routing problem: latest advances and new challenges. Springer, 199-220.

Lau, H. C., Sim, M. and Teo, K. M. (2003). Vehicle routing problem with time windows and a limited number of vehicles. *European Journal of Operational Research* **148:** 559-569.

LAWMA, (2014). Inside LAWMA: Dumpsite. Retrieved from *www.lawma.gov.ng.*

Le Bouthillier, A., and Crainic, T. G. (2005). A cooperative parallel meta-heuristic for the vehicle routing problem with time windows. *Computers and Operations Research* **32(7):** 1685-1708.

Lei, H., Laporte, G., and Guo, B.(2011). The capacitated vehicle routing problem with stochastic demands and time windows. *Computers and Operations Research* **38(12):** 1775-1783.

Lenstra, J. K. and Rinnooy Kan, A. H. G. (1981). Complexity of vehicle routing and scheduling problems. *Networks* **11(2):** 221-227.

Lewis, H. F. and Sexton, T. R. (2007). On the tour partitioning heuristic for the unit demand capacitated vehicle routing problem. *Operations Research Letters* **35:** 773-781.

Liao, T. Y. and Hu, T. Y. (2011). An object oriented evaluation framework for dynamic vehicle routing problems under real-time information. *Expert Systems with Applications* **38:** 12548-12558.

Li, J. Q., Mirchandani, P. B. and Borenstein, D. (2009a). A lagrangian heuristic for real-time vehicle rescheduling problem. *Transportation Research Part E: Logistics and Transportation Review* **45(3):** 419-433.

Lin, S. -W., Lee, Z. -J., Ying, K. -C. and Lee, C. -Y. (2009). Applying hybrid meta-heuristics for capacitated vehicle routing problem. *Expert systems with Applications* **36:** 1505-1512.

Litvinchev, I., Mata, M., Rangel, S. and Saucedo, J. (2010). Lagrangian heuristic for a class of the generalized assignment problems. *Computers and Mathematics with Applications* **60(4):** 1115-1123.

Liu, F. H. and Shen, S. Y. (1999). The fleet size and mix vehicle routing problem with time windows. *Journal of Operational Research Society* **50(7):** 721-732.

Liu, R., Jiang, Z., Fung, R. Y. K., Chen, F. and Liu, X. (2010). An effective memetic algorithm for the cumulative capacitated vehicles routing problem. *Computers and Operations Research* **37:** 1877-1885.

Lorena, L. A. N. and Senne, E. L. F. (2004). A column generation approach to capacitated p-median problems. *Computers and Operations Research* **31:** 863-876.

Lorini, S., Potvin, J. Y. and Zufferey, N. (2011). Online vehicle routing and scheduling with dynamic travel times. *Computers and Operations Research* **38(7):** 1086-1090.

Lund, K., Madsen, O. B. G. and Rygaard, J. M. (1996). Vehicle routing problem with varying degrees of dynamism. Technical Report, IMM, The Department of Mathematical Modeling, Technical University of Denmark, Lyngby, Denmark.

Luo, Z., Qin, H., Che, C. and Lim, A. (2015). On service consistency in multi-period vehicle route. *European Journal of Operational Research* **243:** 731-744.

Lysgaard, J. (2010). The pyramidal capacitated vehicle routing problem. *European Journal of Operational Research* **205:** 59-64.

Lysgaard, J. and Wohlk, S.(2014). A branch-and-cut-and-price algorithm for the cumulative capacitated vehicle routing problem. *European Journal of Operational Research* **236(3):** 800-810.

Marinakis, Y.(2012). Multiple phase neighborhood search-GRASP for capacitated vehicle routing problem. *Expert Systems with Applications* **39:** 6807-6815.

Marti, R., Laguna, M. and Glover, F. (2006). Principles of scatter search. *European Journal of Operational Research* **169:** 359-372.

Mavrovouniotis, M. and Ynag, S. (2015). Ant algorithms with immigrants schemes for the dynamic vehicle routing problem. *Information Sciences* **294:** 456-477.

Mazzeo, S., and Loiseau, I. (2004). An ant colony algorithm for the capacitated vehicle routing problem. *Electronic notes in discrete mathematics* **18:** 181-186.

Mazzola, J. B., and Neebe, A. W. (1999). Lagrangian-relaxation-based solution procedures for a multiproduct capacitated facility location problem with choice of facility type. *European Journal of Operational Research* **115(2):** 285-299.

Mehrjerdi, Y. Z., and Nadizadeh, A. (2013). Using greedy clustering method to solve capacitated location-routing problem with fuzzy demands. *European Journal of Operational Research* **229(1):** 75-84.

Merkle, D. and Middendorf, M. (2005). Swarm intelligence. In: B. E. Burke & G. Kendall (Eds). Search methodology: introductory tutorials in optimization and decision support techniques. Springer Science + Business Media, LLC, ISBN: 0-0387-23460-8.

Mester, D, Braysy, O. and Dullaert, W. (2005). A multi-parametric evolution strategies algorithm for capacitated vehicle routing problem. Working paper, University of Haifa, Israel.

Mester, D. and Braysy, O. (2007). Active-guided evolution strategies for large-scale capacitated vehicle routing problems. *Computers and Operations Research* **34:** 2964-2975.

Michallet, J., Prins, C., Amodeo, L., Yalaoui, F. and Vitry, G. (2014). Multi-start iterated local search for the periodic vehicle routing problem with time windows and spread constraints on services. *Computers and Operations Research* **41:** 196-207.

Mitchell, M. (1995). Genetic algorithm: an overview. *Complexity* **1(1):** 31-39.

Mladenović, N., Brimberg, J., Hansen, P. and Moreno-Perez, J. (2007). The p-median problem: a survey of metaheuristic approaches. *European Journal of Operational Research* **179:** 927-939.

Mu, Q., Fu, Z., Lysgaard, J. and Eglese, E. (2011). Disruption management of the vehicle routing problem with vehicle breakdown. *Journal of the Operational Research Society* **64(2):** 742-749.

Nagata, Y. and Braysy, O. (2009). A powerful route optimization heuristic for the vehicle routing problem with time windows. *Operations Research Letters* **37:** 333-338.

Nagata, Y., Braysy, O. and Dullaert, W. (2010). A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers and Operations Research* **37:** 724-737.

Nazif, H. and Lee, L. S. (2012). Optimised crossover genetic algorithm for capacitated vehicle routing problem. *Applied Mathematical Modelling* **36(5):** 2110-2117.

Nemhauser, G. L. and Wolsey, L. A. (1988). Integer and combinatorial optimization. John Wiley Sons, Inc.

Nezhad, A. M., Manzour, H. and Salhi, S. (2013). Lagrangian relaxation heuristics for the uncapacitated single source multi-product facility location problem. *International Journal of Production Economics* **145(2):** 713-723.

Ngueveu, S. U., Prins, C. and Wolfler Calvo, R. (2010). An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Computers and Operations Research* **37(11)** 1877-1885.

Nguyen, P. K., Crainic, T. G. and Toulouse, M. (2011). A hybrid genetic algorithm for periodic vehicle routing problems with time windows. CIRRELT-2011-25.

Nguyen, P.K., Crainic, T.G. and Toulouse, M. (2013). A tabu search for time-dependent multi-zone-multi-trip vehicle routing problem with time windows. *European Journal of Operations Research* **231:** 43-56.

Oduro-Kwarteng, S. (2011). Private sector involvement in urban solid waste collection: performance, capacity and regulation in five cities in Ghana. Taylor and Francis, USA.

Ogryczak, W. (2000). Inequality measures and equitable approaches to location problems. *European Journal of Operational Research* **122:** 347-391.

Ogwueleka, T. C. (2009). Municipal Solid Waste Characteristics and management in Nigeria. *Iran. J. Environ. Health Sci. Eng.* **6(3):** 173-180.

Olusanya, M. O., Arasomwan, M. A. and Adewumi, A. O. (2015). Particle swarm optimization algorithm for optimizing assignment of blood in blood banking system. *Computational and Mathematical Methods in Medicine* **2015:** 1-12.

Ombuki-Berman, B. M., Runka, A. and Hanshar, F. T. (2007). Waste collection vehicle routing problem with time windows using multiobjective genetic algorithms. Brock University, Technical Report CS-07-04, 91-97.

Pang, K. W. (2011). An adaptive parallel route construction heuristic for the VRP with time windows constraints. *Expert Systems with Applications* **38:** 11939-11946.

Parsopoulos, K. E. and Vrahatis, M. N. (2002). Particle Swarm Optimization Method in Multiobjective Problems. In: *Proceedings of ACM 2002 Symposium on Applied Computing SAC* **2002:** 603-607.

Pillac, V., Gendreau, M., Gueret, C. and Medaglia, A. L. (2011). A review of dynamic vehicle routing problems. Inter University Research Centre on Enterprise Networks, Logistics and Transportation, CIRRELT-2011-62.

Pillac, V., Gueret, C. and Medaglia, A. L. (2012). An event-driven optimization framework for dynamic vehicle routing. *Decision Support Systems* **54:** 414-423.

Pisinger, D. (1995). Algorithms for knapsack problems. Ph.D Thesis, Dept. of Computer Science, University of Copenhagen, Denmark.

Pisinger, D. and Ropke, S. (2010). Large neighborhood search. In *Handbook of metaheuristics*. Springer US, 399-419.

Polyak, B. T. (1969). Minimization of unsmooth functionals. *USSR Computational Mathematics and Mathematical Physics* **9:** 14-29.

Potvin, J. Y., Zu, Y. and Benyahia, I. (2006). Vehicle routing and scheduling with dynamic travel times. *Computers and Operations Research* **33:** 1129-1137.

Psaraftis, H. N. (1980). A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science* **14(2):** 130-154.

Qureshi, A. G., Tanigushi, E. and Yamada, T. (2009). An exact solution approach for vehicle routing and scheduling problems with soft time windows. *Transportation Research Part E* **45:** 960-977.

Qureshi, A. G., Taniguchi, E. and Yamada, T. (2010). Exact solution for the vehicle routing problem with semi soft time windows and its application. *Procedia - Social and Behavioral Sciences* **2(3):** 5931-5943.

Qureshi, A. G., Taniguchi, E. and Yamada, T. (2012). A microsimulation based analysis of exact solution of dynamic vehicle routing with soft time windows. *Procedia - Social and Behavioural Sciences* **39:** 205-216.

Ribeiro, G. M. and Laporte, G. (2012). An adaptive large neighborhood search heuristic for the cumulative capacitated vehicles routing problem. *Computers and Operations Research* **39:** 1419-1431.

Rivera, J. C., Murat Afsar, H. and Prins, C. (2016). Mathematical formulations and exact algorithm for the multitrip cumulative capacitated single-vehicle routing problem. *European Journal of Operational Research* **249(1):** 93-104.

Rodriguez, A. and Ruiz, R. (2012). A study on the effect of the asymmetric on real capacitated vehicle routing problems. *Computers and Operations Research* **39:** 2142-2151.

Ropke, S. and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science* **40(4):** 455-472.

Russell, R. A. (1995). Hybrid heuristics for the vehicle routing problem with time windows. *Transportation Science* **29:** 156-166.

Santosa, B. and Kresna, I. G. N. A. (2015). Simulated annealing to solve single stage capacitated warehouse location problem. *Procedia Manufacturing* **4:** 62-70.

Sarker, R. A. and Newton, C. S. (2008). Optimization modelling: A practical approach. Taylor and Francis, NW, USA.

Schyns, M. (2015). An ant colony system for responsive dynamic vehicle routing. *European Journal of Operational Research* **245:** 704-718.

Sean, L. (2013). Essentials of metaheuristics. Lulu, second edition, available at http://cs.gmu.edu/ sean/book/metaheuristics/.

Sharma, K., Chhamunya, V., Gupta, P. C., Sharma, H. and Bansal, J. C. (2015). Fitness based particle swarm optimization. *International Journal of System Assurance Engineering and Management* **6(3):** 319-329.

Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In CP-98 (Fourth International Conference on Principles and Practice of Constraint Programming), volume 1520 of Lecture Notes in Computer Science, 417-431.

Solomon, M. M. (1987). Algorithms for vehicle routing and scheduling problems with time window constraints. *Operations Research* **35:** 254-265.

Solomon, M. M. and Desrosiers, J. (1988). Survey paper - time window constrained routing and scheduling problems. *Transportation Science* **22(1):** 1-13.

Sorensen, K. and Schittekat, P. (2013). Statistical analysis of distance-based path relinking for the capacitated vehicle routing problem. *Computers and Operations Research* **40:** 3197-3205.

Stanojevic, M., Stanojevic, B. and Vujosevic, M. (2013). Enhanced savings calculation and its applications for solving capacitated vehicle routing problem. *Applied Mathematics and Computation* **219:** 10302-10312.

Sule, D. R. (2001). Logistics of facility location and allocation. Marcel Dekker Inc., New York.

Szeto, W. Y., Wu, Y. and Ho, S.C. (2011). An artificial bee colony algorithm for the capacitated vehicle routing problems. *European Journal of Operational Research* **215:** 126-135.

Taillard, E. (1993). Parallel iterative search methods for vehicle routing problems. /em Networks **23:** 661-673.

Taillard, E. (1995). A heuristic column generation method for the heterogeneous fleet vehicle routing problem. *RAIRO* **33(1):** 1-14.

Tan, C. C. R., and Beasley, J. E. (1984). A heuristic algorithm for the period vehicle routing problem. *Omega* **12(5):** 497-504.

Tarakkoli-Moghaddam, R., Safali, N. and Gholipour, Y. (2006). A hybrid simulated annealing for capacitated vehicle routing problems with the independent route length. *Applied Mathematics and Computation* **176:** 445-454.

Tarakkoli-Mohghaddam, R., Safali, N., Kah, M.M.O. and Rabbani, M. (2007). A new capacitated vehicle routing problem with split service for minimizing float cost by simulated Annealing. *Journal of the Franklin Institute* **344:** 406-425.

Tchobanoglous, G. and Kreith, F. (2002). Handbook of Solid Waste Management. McGraw-Hill, USA, Second Edition.

Teixeira, J., Antunes, A. P. and de Sousa, J. P. (2004). Recyclable waste collection planning-a case study. *European Journal of Operational Research* **158(3):** 543-554.

Tlili, T., Faiz, S. and Krichen, S. (2014). An hybrid metaheuristic for the distance-constrained capacitated vehicle routing problem. *Procedia-Social and Behavioral Sciences* **109:** 779-783.

Thomas, B. W. (2007). Waiting strategies for anticipating service requests from known customer locations. *Transportation Science* **41(3):** 319-331.

Toth, T. and Vigo, D. (2002). Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics* **123:** 427-512.

Tragantalerngsak, S., Holt, J. and Rnnqvist, M. (2000). An exact method for the two-echelon, single-source, capacitated facility location problem. *European Journal of Operational Research* **123(3):** 473-489.

Tralhao, L., Coutinho-Rodrigues, J. and Alćada-Almeida, L. (2010). A multiobjective modelling approach to locate multi-compartment containers for urban-sorted waste. *Waste Management* **30:** 2418-2429.

Ursani, Z., Essam, D., Cornforth, D. and Stocker, R. (2011). Localized genetic algorithm for vehicle routing problem with time windows. *Applied Soft Computing* **11:** 5375-5390.

US EPA (2008). Municipal solid waste generation, recycling and disposal in the United States, Facts and Figures for 2007, EPA-530-R-08-0 1 0.

Uster, H. and Love, R. F. (2003). Formulation of confidence intervals for estimated actual intervals. *European Journal of Operational Research* **151:** 586-601.

Van Woensel, T., Kerbache, L., Peremans, H. and Vandaele, N. (2008). Vehicle routing with dynamic travel times: a queuing approach. *European Journal of Operational Research* **186:** 990-1007.

Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N. and Rei, W. (2012). A hybrid genetic algorithm for multi-depot and periodic vehicle routing problems. *Operations Research* **60(3):** 611-624.

Wahab, W. B. (2012). Solid waste: issue and challenges in urban context. Retrieved from http://start.org/download/2012/ai-irdr/data/2-1-solidwaste.pdf, March 2015.

Wang, C. H. and Lu, J. Z. (2009). A hybrid genetic algorithm that optimizes capacitated vehicle routing problems. *Expert systems with applications* **36:** 2921-2936.

Wen, M., Cordeau, J. F., Laporte, G. and Larsen, J. (2010). The dynamic multi-period vehicle routing problem. *Computers and Operations Research* **37:** 1615-1623.

Wikipedia (2013). Eti-Osa. Retrieved from https://en.wikipedia.org/wiki/Eti-Osa.

Wilson, N. H., and Colvin, N. J. (1977). Computer control of the Rochester dial-a-ride system. Massachusetts Institute of Technology, Center for Transportation Studies, USA.

Wohlgemuth, S., Oloruntoba, R. and Clausen, U. (2012). Dynamic vehicle routing with anticipation in disaster relief. *Socio-Economic Planning Sciences* **46:** 261-271.

Wolsey, L. A. (1998). Integer programming. John Wiley Sons, Inc., USA.

Xiao, Y., Zhao, Q., Kaku, I. and Xiu, Y. (2012). Development of a fuel consumption

optimization model for the capacitated vehicle routing problem. *Computers and Operations Research* **39:** 1419-14131.

Yamashita, D. S., Armento, V. A. and Laguna, M. (2006). Scatter search for project scheduling with resource availability cost. *European Journal of Operational Research* **169:** 623-637.

Yang, Z., Chen, H. and Chu, F. (2011). A Lagrangian relaxation approach for a large scale new variant of capacitated clustering problem. *Computers and Industrial Engineering* **61(2):** 430-435.

Yu, B. and Yang, Z. Z. (2011). An ant colony optimization model: the periodic vehicle routing problem with time windows. *Transportation Research Part E* **47:** 166-181.

Yurtjuran, A. and Einel, E. (2010). A new hybrid electromagnetism-like algorithm for capacitated vehicle routing problems. *Expert systems with applications* **37:** 3427-3433.

Zhang, Z., Che, O., Cheang, B., Lim, A. and Qin, H. (2013). A memetic algorithm for the multi-period vehicle routing problem with profit. *European Journal of Operational Research* **229:** 573-584.

APPENDIX A

SAMPLE AMPL CODES FOR MODEL I

1.1 Model Building Code

```
set cust; # set of customers
set POT;   # set of potential collection facilities
set cont;  # set of different containers
param D >= 0; # threshold distance
param c {t in cont}; # capacity of each container type
param mu; # a large constant
param n {t in cont} integer >0;
param Beta {t in cont, i in cust};
param d {i in cust, k in POT}; #
var BUILT{k in POT} binary; #
var ASSIGN{i in cust, k in POT} binary;
var ALLOCATE{t in cont, k in POT} integer >= 0;
minimize total_facility:
sum {k in POT} BUILT[k];
subject to constraint_1 {i in cust}:
sum {k in POT} ASSIGN[i,k] = 1;
subject to constraint_2 {t in cont}:
sum {k in POT} ALLOCATE[t,k] <= n [t];
subject to constraint_3 {k in POT}:
sum {i in cust} sum {t in cont} Beta[t,i]
* ASSIGN[i,k] <= sum {t in cont} c[t] * ALLOCATE[t,k];
subject to constraint_4 {k in POT: mu >= card(cont)}:
mu * BUILT[k] >= sum {t in cont} ALLOCATE[t,k];
subject to constraint_5 {i in cust, k in POT}:
(d[i,k] * ASSIGN[i,k]) <= D;
subject to capacity {i in cust}:
sum {t in cont} n[t]*c[t] >= sum {t in cont} Beta[t,i];
```

1.2 Subgradient Algorithm Run Code for Model I

```
# ----------------------------------------
# LAGRANGIAN RELAXATION FOR MODEL ONE
# ----------------------------------------
reset;
printf "\nLP RELAXATION\n\n";
model slr.mod;
data matrix100.dat;
#data Model1.dat
option omit_zero_rows 1;
option display_eps .000001;
option solution_round 8;
option presolve 0;
#option cplex_option 'presolve_eps >= 3.6e-08';
```

```
option solver cplex;
option solver_msg 0;
#option cplex_options 'mipdisplay=1';
option relax_integrality 1;
objective total_facility;
solve;
param LB; param UB;
#let LB := 0;
let UB := 100;
let LB := total_facility.val;
#let UB := 100;
option relax_integrality 1;
#problem LowerBound: Lagrangian;
#problem UpperBound: total_facility;
problem LowerBound: ALLOCATE, constraint_2, constraint_3,
constraint_4, constraint_5, Lagrangian;
#problem LowerBound: constraint_2, constraint_3,
constraint_4, constraint_5, Lagrangian;
#problem UpperBound: constraint_1, constraint_2, constraint_3,
constraint_4, constraint_5, ALLOCATE, ASSIGN, BUILT, total_facility;
#problem LowerBound: constraint_2, constraint_3,
constraint_4, constraint_5, Lagrangian;
problem UpperBound:  ALLOCATE, BUILT, constraint_2,
constraint_3, constraint_4, constraint_5, total_facility;
#problem UpperBound:  ALLOCATE, constraint_1, constraint_2,
constraint_3, constraint_4, constraint_5, total_facility;
let {i in cust} mult[i] := 0;
param slack {cust};
param scale  default 1;
param norm;
param step;
param same  default 0;
param same_limit := 3;
param iter_limit := 10;
param LBlog {0..iter_limit};  let LBlog[0] := LB;
param UBlog {0..iter_limit};  let UBlog[0] := UB;
param scalelog {1..iter_limit};
param steplog {1..iter_limit};
for {m in 1..iter_limit} { printf "\nITERATION %d\n\n", m;
   solve LowerBound;
   let {i in cust} slack[i] := sum {k in POT} ASSIGN[i,k] - 1;
   if Lagrangian > LB + 0.000001 then {
      let LB := Lagrangian;
      let same := 0; }
   else let same := same + 1;
   if same = same_limit then {
      let scale := scale / 2;
      let same := 0;
      };
   let norm := sum {i in cust} slack[i]^2;
   let step := scale * (UB - Lagrangian) / (norm+1);

   let {i in cust} mult[i] := mult[i] less step * slack[i];
```

```
     if sum {t in cont} n[t]*c[t]
     >= sum {t in cont}sum {i in cust} Beta[t,i] then {
        solve UpperBound;
      let UB := min (UB, total_facility);
      }
#- 1e-8
   let LBlog[m] := LB;
   let UBlog[m] := UB;
   let scalelog[m] := scale;
   let steplog[m] := step;
   }
printf "\n\n";
display LBlog, UBlog, scalelog, steplog;
```

1.3 Sample of a Pseudo Data File

```
data;
set cust := C1 C2 C3 C4 C5 C1 C2 C3 C4 C5 C1 C2 C3 C4 C5 C1 C2 C3 C4 C5;
set POT :=  P1 P2 P3 P4 P5;
set cont := X Y Z;
param D := 150;
param mu := 10;
param c :=
X 150
Y 150
Z 150;
param n :=
X 15
Y 30
Z 21;
param d:  P1 P2 P3 P4 P5:=
C1 45 180 200 48 91
C2 100 120 110 115 86
C3 112 170 181 136 144
C4 170 130 108 116 70
C5 135 250 218 210 181;
param Beta:  C1 C2 C3 C4 C5:=
X 10 40 50 10 70
Y 50 36 37 80 20
Z 15 45 10 20 51;
end;
```

APPENDIX B

SAMPLE OF AMPL CODES FOR MODEL II

2.1 Model Building Code

```
#The vehicle routing problem with time windows and road attribute
set cust;
set links within {cust cross cust};
param XCoord {f in cust};
param YCoord {k in cust};
param twlb {f in cust}; # lower bound of the time window
param twub {f in cust}; # upper bound of the time window
param service {f in cust}; # service time at customer f
param demand {f in cust}; # demand of customer f
param travel_time {(f,k) in links};
param distance {(f,k) in links};
param C > 0; # vehicle capacity
param M; # a large constant
param roadatt {(f,k) in links};
var route {(f,k) in links} binary;
var arrive {f in cust} >=0;
var load {f in cust} >=0;
minimize total_cost:
sum {(f,k) in links} distance[f,k]*route[f,k];
subject to constraint_1 {f in cust}:
sum {k in cust}route[f,k] = 1;
subject to constraint_2 {f in cust}:
sum {k in cust}route[f,k] - sum {k in cust}route[k,f] = 0;
subject to constraint_3 {(f,k) in links}:
arrive[f] + service[f] + travel_time[f,k] <= arrive[k] + (1 - route[f,k])*M;
subject to constraint_4 {(f,k) in links}:
load[f] + demand[f] <= load[k] + (1 - route[f,k])*M;
subject to constraint_5 {f in cust}:
twlb[f] <= arrive[f] <= twub[f];
subject to constraint_9 {f in cust}:
0 <= load[f] <= C;
```

2.2 Sample of a Data File

```
data;

set links :=
(0, *)  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(1, *)  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(2, *)  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(3, *)  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(4, *)  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(5, *)  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
```

```
(6, *)  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(7, *)  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(8, *)  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(9, *)  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(10, *) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(11, *) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(12, *) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(13, *) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(14, *) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(15, *) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(16, *) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(17, *) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(18, *) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(19, *) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(20, *) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(21, *) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(22, *) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(23, *) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(24, *) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
(25, *) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25;

param C := 200;
param M :=1000;
param:
cust:     XCoord     YCoord    demand      twlb      twub      service:=
    0      40         50          0         0      1236          0
    1      45         68         10       912       967         90
    2      45         70         30       825       870         90
    3      42         66         10        65       146         90
    4      42         68         10       727       782         90
    5      42         65         10        15        67         90
    6      40         69         20       621       702         90
    7      40         66         20       170       225         90
    8      38         68         20       255       324         90
    9      38         70         10       534       605         90
   10      35         66         10       357       410         90
   11      35         69         10       448       505         90
   12      25         85         20       652       721         90
   13      22         75         30        30        92         90
   14      22         85         10       567       620         90
   15      20         80         40       384       429         90
   16      20         85         40       475       528         90
   17      18         75         20        99       148         90
   18      15         75         20       179       254         90
   19      15         80         10       278       345         90
   20      30         50         10        10        73         90
   21      30         52         20       914       965         90
   22      28         52         20       812       883         90
   23      28         55         10       732       777         90
   24      25         50         10        65       144         90
   25      25         52         40       169       224         90;
```

APPENDIX C

SAMPLE OF OUTPUT FROM IMPLEMENTATION OF MODEL I

3.1 Output for Location-handling Variables

```
<variables>
 <variable name="BUILT(A001)" index="0" value="0"/>
 <variable name="BUILT(A002)" index="1" value="1"/>
 <variable name="BUILT(A003)" index="2" value="0"/>
 <variable name="BUILT(A004)" index="3" value="0"/>
 <variable name="BUILT(A005)" index="4" value="1"/>
 <variable name="BUILT(A006)" index="5" value="0"/>
 <variable name="BUILT(A007)" index="6" value="0"/>
 <variable name="BUILT(A008)" index="7" value="1"/>
 <variable name="BUILT(A009)" index="8" value="0"/>
 <variable name="BUILT(A010)" index="9" value="0"/>
 <variable name="BUILT(A011)" index="10" value="0"/>
 <variable name="BUILT(A012)" index="11" value="0"/>
 <variable name="BUILT(A013)" index="12" value="0"/>
 <variable name="BUILT(A014)" index="13" value="0"/>
 <variable name="BUILT(A015)" index="14" value="0"/>
 <variable name="BUILT(A016)" index="15" value="0"/>
 <variable name="BUILT(A017)" index="16" value="0"/>
 <variable name="BUILT(A018)" index="17" value="1"/>
 <variable name="BUILT(A019)" index="18" value="0"/>
 <variable name="BUILT(A020)" index="19" value="0"/>
 <variable name="BUILT(A021)" index="20" value="0"/>
 <variable name="BUILT(A022)" index="21" value="0"/>
 <variable name="BUILT(A023)" index="22" value="1"/>
 <variable name="BUILT(A024)" index="23" value="0"/>
 <variable name="BUILT(A025)" index="24" value="0"/>
 <variable name="BUILT(A026)" index="25" value="0"/>
 <variable name="BUILT(A027)" index="26" value="0"/>
 <variable name="BUILT(A028)" index="27" value="0"/>
 <variable name="BUILT(A029)" index="28" value="1"/>
 <variable name="BUILT(A030)" index="29" value="0"/>
 <variable name="BUILT(A031)" index="30" value="0"/>
 <variable name="BUILT(A032)" index="31" value="0"/>
 <variable name="BUILT(A033)" index="32" value="0"/>
 <variable name="BUILT(A034)" index="33" value="0"/>
 <variable name="BUILT(A035)" index="34" value="0"/>
 <variable name="BUILT(A036)" index="35" value="0"/>
 <variable name="BUILT(A037)" index="36" value="1"/>
 <variable name="BUILT(A038)" index="37" value="1"/>
 <variable name="BUILT(A039)" index="38" value="1"/>
 <variable name="BUILT(A040)" index="39" value="0"/>
 <variable name="BUILT(A041)" index="40" value="1"/>
 <variable name="BUILT(A042)" index="41" value="0"/>
 <variable name="BUILT(A043)" index="42" value="0"/>
```

<variable name="BUILT(A044)" index="43" value="0"/>
<variable name="BUILT(A045)" index="44" value="0"/>
<variable name="BUILT(A046)" index="45" value="0"/>
<variable name="BUILT(A047)" index="46" value="1"/>
<variable name="BUILT(A048)" index="47" value="0"/>
<variable name="BUILT(A049)" index="48" value="0"/>
<variable name="BUILT(A050)" index="49" value="1"/>
<variable name="BUILT(A051)" index="50" value="0"/>
<variable name="BUILT(A052)" index="51" value="0"/>
<variable name="BUILT(A053)" index="52" value="1"/>
<variable name="BUILT(A054)" index="53" value="1"/>
<variable name="BUILT(A055)" index="54" value="0"/>
<variable name="BUILT(A056)" index="55" value="0"/>
<variable name="BUILT(A057)" index="56" value="0"/>
<variable name="BUILT(A058)" index="57" value="0"/>
<variable name="BUILT(A059)" index="58" value="0"/>
<variable name="BUILT(A060)" index="59" value="0"/>
<variable name="BUILT(A061)" index="60" value="0"/>
<variable name="BUILT(A062)" index="61" value="1"/>
<variable name="BUILT(A063)" index="62" value="0"/>
<variable name="BUILT(A064)" index="63" value="0"/>
<variable name="BUILT(A065)" index="64" value="0"/>
<variable name="BUILT(A066)" index="65" value="0"/>
<variable name="BUILT(A067)" index="66" value="1"/>
<variable name="BUILT(A068)" index="67" value="0"/>
<variable name="BUILT(A069)" index="68" value="1"/>
<variable name="BUILT(A070)" index="69" value="1"/>
<variable name="BUILT(A071)" index="70" value="0"/>
<variable name="BUILT(A072)" index="71" value="0"/>
<variable name="BUILT(A073)" index="72" value="0"/>
<variable name="BUILT(A074)" index="73" value="0"/>
<variable name="BUILT(A075)" index="74" value="0"/>
<variable name="BUILT(A076)" index="75" value="0"/>
<variable name="BUILT(A077)" index="76" value="0"/>
<variable name="BUILT(A078)" index="77" value="0"/>
<variable name="BUILT(A079)" index="78" value="0"/>
<variable name="BUILT(A080)" index="79" value="1"/>
<variable name="BUILT(A081)" index="80" value="1"/>
<variable name="BUILT(A082)" index="81" value="0"/>
<variable name="BUILT(A083)" index="82" value="0"/>
<variable name="BUILT(A084)" index="83" value="0"/>
<variable name="BUILT(A085)" index="84" value="0"/>
<variable name="BUILT(A086)" index="85" value="0"/>
<variable name="BUILT(A087)" index="86" value="0"/>
<variable name="BUILT(A088)" index="87" value="1"/>
<variable name="BUILT(A089)" index="88" value="0"/>
<variable name="BUILT(A090)" index="89" value="0"/>
<variable name="BUILT(A091)" index="90" value="0"/>
<variable name="BUILT(A092)" index="91" value="0"/>
<variable name="BUILT(A093)" index="92" value="1"/>
<variable name="BUILT(A094)" index="93" value="0"/>
<variable name="BUILT(A095)" index="94" value="1"/>
<variable name="BUILT(A096)" index="95" value="0"/>

<variable name="BUILT(A097)" index="96" value="0"/>
<variable name="BUILT(A098)" index="97" value="1"/>
<variable name="BUILT(A099)" index="98" value="0"/>
<variable name="BUILT(A100)" index="99" value="0"/>
<variable name="BUILT(B001)" index="100" value="0"/>
<variable name="BUILT(B002)" index="101" value="0"/>
<variable name="BUILT(B003)" index="102" value="0"/>
<variable name="BUILT(B004)" index="103" value="1"/>
<variable name="BUILT(B005)" index="104" value="0"/>
<variable name="BUILT(B006)" index="105" value="0"/>
<variable name="BUILT(B007)" index="106" value="0"/>
<variable name="BUILT(B008)" index="107" value="0"/>
<variable name="BUILT(B009)" index="108" value="1"/>
<variable name="BUILT(B010)" index="109" value="1"/>
<variable name="BUILT(B011)" index="110" value="0"/>
<variable name="BUILT(B012)" index="111" value="0"/>
<variable name="BUILT(B013)" index="112" value="0"/>
<variable name="BUILT(B014)" index="113" value="0"/>
<variable name="BUILT(B015)" index="114" value="0"/>
<variable name="BUILT(B016)" index="115" value="0"/>
<variable name="BUILT(B017)" index="116" value="0"/>
<variable name="BUILT(B018)" index="117" value="1"/>
<variable name="BUILT(B019)" index="118" value="0"/>
<variable name="BUILT(B020)" index="119" value="0"/>
<variable name="BUILT(B021)" index="120" value="0"/>
<variable name="BUILT(B022)" index="121" value="0"/>
<variable name="BUILT(B023)" index="122" value="0"/>
<variable name="BUILT(B024)" index="123" value="0"/>
<variable name="BUILT(B025)" index="124" value="0"/>
<variable name="BUILT(B026)" index="125" value="0"/>
<variable name="BUILT(B027)" index="126" value="0"/>
<variable name="BUILT(B028)" index="127" value="1"/>
<variable name="BUILT(B029)" index="128" value="1"/>
<variable name="BUILT(B030)" index="129" value="0"/>
<variable name="BUILT(B031)" index="130" value="1"/>
<variable name="BUILT(B032)" index="131" value="0"/>
<variable name="BUILT(B033)" index="132" value="0"/>
<variable name="BUILT(B034)" index="133" value="0"/>
<variable name="BUILT(B035)" index="134" value="0"/>
<variable name="BUILT(B036)" index="135" value="0"/>
<variable name="BUILT(B037)" index="136" value="0"/>
<variable name="BUILT(B038)" index="137" value="0"/>
<variable name="BUILT(B039)" index="138" value="0"/>
<variable name="BUILT(B040)" index="139" value="0"/>
<variable name="BUILT(B041)" index="140" value="1"/>
<variable name="BUILT(B042)" index="141" value="0"/>
<variable name="BUILT(B043)" index="142" value="0"/>
<variable name="BUILT(B044)" index="143" value="0"/>
<variable name="BUILT(B045)" index="144" value="0"/>
<variable name="BUILT(B046)" index="145" value="0"/>
<variable name="BUILT(B047)" index="146" value="0"/>
<variable name="BUILT(B048)" index="147" value="0"/>
<variable name="BUILT(B049)" index="148" value="0"/>

<variable name="BUILT(B050)" index="149" value="0"/>
<variable name="BUILT(B051)" index="150" value="0"/>
<variable name="BUILT(B052)" index="151" value="0"/>
<variable name="BUILT(B053)" index="152" value="0"/>
<variable name="BUILT(B054)" index="153" value="0"/>
<variable name="BUILT(B055)" index="154" value="1"/>
<variable name="BUILT(B056)" index="155" value="1"/>
<variable name="BUILT(B057)" index="156" value="0"/>
<variable name="BUILT(B058)" index="157" value="0"/>
<variable name="BUILT(B059)" index="158" value="0"/>
<variable name="BUILT(B060)" index="159" value="0"/>
<variable name="BUILT(B061)" index="160" value="1"/>
<variable name="BUILT(B062)" index="161" value="0"/>
<variable name="BUILT(B063)" index="162" value="0"/>
<variable name="BUILT(B064)" index="163" value="0"/>
<variable name="BUILT(B065)" index="164" value="0"/>
<variable name="BUILT(B066)" index="165" value="0"/>
<variable name="BUILT(B067)" index="166" value="0"/>
<variable name="BUILT(B068)" index="167" value="0"/>
<variable name="BUILT(B069)" index="168" value="1"/>
<variable name="BUILT(B070)" index="169" value="0"/>
<variable name="BUILT(B071)" index="170" value="0"/>
<variable name="BUILT(B072)" index="171" value="0"/>
<variable name="BUILT(B073)" index="172" value="0"/>
<variable name="BUILT(B074)" index="173" value="0"/>
<variable name="BUILT(B075)" index="174" value="0"/>
<variable name="BUILT(B076)" index="175" value="1"/>
<variable name="BUILT(B077)" index="176" value="0"/>
<variable name="BUILT(B078)" index="177" value="0"/>
<variable name="BUILT(B079)" index="178" value="0"/>
<variable name="BUILT(B080)" index="179" value="1"/>
<variable name="BUILT(B081)" index="180" value="0"/>
<variable name="BUILT(B082)" index="181" value="0"/>
<variable name="BUILT(B083)" index="182" value="0"/>
<variable name="BUILT(B084)" index="183" value="0"/>
<variable name="BUILT(B085)" index="184" value="0"/>
<variable name="BUILT(B086)" index="185" value="0"/>
<variable name="BUILT(B087)" index="186" value="0"/>
<variable name="BUILT(B088)" index="187" value="0"/>
<variable name="BUILT(B089)" index="188" value="1"/>
<variable name="BUILT(B090)" index="189" value="0"/>
<variable name="BUILT(B091)" index="190" value="0"/>
<variable name="BUILT(B092)" index="191" value="0"/>
<variable name="BUILT(B093)" index="192" value="0"/>
<variable name="BUILT(B094)" index="193" value="0"/>
<variable name="BUILT(B095)" index="194" value="1"/>
<variable name="BUILT(B096)" index="195" value="0"/>
<variable name="BUILT(B097)" index="196" value="1"/>
<variable name="BUILT(B098)" index="197" value="0"/>
<variable name="BUILT(B099)" index="198" value="0"/>
<variable name="BUILT(B100)" index="199" value="0"/>
<variable name="BUILT(C001)" index="200" value="0"/>
<variable name="BUILT(C002)" index="201" value="0"/>

<variable name="BUILT(C003)" index="202" value="0"/>
<variable name="BUILT(C004)" index="203" value="0"/>
<variable name="BUILT(C005)" index="204" value="0"/>
<variable name="BUILT(C006)" index="205" value="0"/>
<variable name="BUILT(C007)" index="206" value="0"/>
<variable name="BUILT(C008)" index="207" value="0"/>
<variable name="BUILT(C009)" index="208" value="0"/>
<variable name="BUILT(C010)" index="209" value="0"/>
<variable name="BUILT(C011)" index="210" value="0"/>
<variable name="BUILT(C012)" index="211" value="0"/>
<variable name="BUILT(C013)" index="212" value="0"/>
<variable name="BUILT(C014)" index="213" value="0"/>
<variable name="BUILT(C015)" index="214" value="0"/>
<variable name="BUILT(C016)" index="215" value="0"/>
<variable name="BUILT(C017)" index="216" value="0"/>
<variable name="BUILT(C018)" index="217" value="0"/>
<variable name="BUILT(C019)" index="218" value="0"/>
<variable name="BUILT(C020)" index="219" value="0"/>
<variable name="BUILT(C021)" index="220" value="0"/>
<variable name="BUILT(C022)" index="221" value="0"/>
<variable name="BUILT(C023)" index="222" value="0"/>
<variable name="BUILT(C024)" index="223" value="0"/>
<variable name="BUILT(C025)" index="224" value="0"/>
<variable name="BUILT(C026)" index="225" value="0"/>
<variable name="BUILT(C027)" index="226" value="0"/>
<variable name="BUILT(C028)" index="227" value="0"/>
<variable name="BUILT(C029)" index="228" value="0"/>
<variable name="BUILT(C030)" index="229" value="0"/>
<variable name="BUILT(C031)" index="230" value="0"/>
<variable name="BUILT(C032)" index="231" value="1"/>
<variable name="BUILT(C033)" index="232" value="0"/>
<variable name="BUILT(C034)" index="233" value="0"/>
<variable name="BUILT(C035)" index="234" value="0"/>
<variable name="BUILT(C036)" index="235" value="0"/>
<variable name="BUILT(C037)" index="236" value="0"/>
<variable name="BUILT(C038)" index="237" value="0"/>
<variable name="BUILT(C039)" index="238" value="0"/>
<variable name="BUILT(C040)" index="239" value="0"/>
<variable name="BUILT(C041)" index="240" value="0"/>
<variable name="BUILT(C042)" index="241" value="0"/>
<variable name="BUILT(C043)" index="242" value="0"/>
<variable name="BUILT(C044)" index="243" value="0"/>
<variable name="BUILT(C045)" index="244" value="0"/>
<variable name="BUILT(C046)" index="245" value="0"/>
<variable name="BUILT(C047)" index="246" value="0"/>
<variable name="BUILT(C048)" index="247" value="0"/>
<variable name="BUILT(C049)" index="248" value="0"/>
<variable name="BUILT(C050)" index="249" value="0"/>
<variable name="BUILT(C051)" index="250" value="1"/>
<variable name="BUILT(C052)" index="251" value="0"/>
<variable name="BUILT(C053)" index="252" value="0"/>
<variable name="BUILT(C054)" index="253" value="0"/>
<variable name="BUILT(C055)" index="254" value="0"/>

```
<variable name="BUILT(C056)" index="255" value="0"/>
<variable name="BUILT(C057)" index="256" value="0"/>
<variable name="BUILT(C058)" index="257" value="1"/>
<variable name="BUILT(C059)" index="258" value="0"/>
<variable name="BUILT(C060)" index="259" value="0"/>
<variable name="BUILT(C061)" index="260" value="1"/>
<variable name="BUILT(C062)" index="261" value="1"/>
<variable name="BUILT(C063)" index="262" value="0"/>
<variable name="BUILT(C064)" index="263" value="0"/>
<variable name="BUILT(C065)" index="264" value="0"/>
<variable name="BUILT(C066)" index="265" value="0"/>
<variable name="BUILT(C067)" index="266" value="0"/>
<variable name="BUILT(C068)" index="267" value="0"/>
<variable name="BUILT(C069)" index="268" value="0"/>
<variable name="BUILT(C070)" index="269" value="0"/>
<variable name="BUILT(C071)" index="270" value="0"/>
<variable name="BUILT(C072)" index="271" value="0"/>
<variable name="BUILT(C073)" index="272" value="0"/>
<variable name="BUILT(C074)" index="273" value="1"/>
<variable name="BUILT(C075)" index="274" value="0"/>
<variable name="BUILT(C076)" index="275" value="0"/>
<variable name="BUILT(C077)" index="276" value="1"/>
<variable name="BUILT(C078)" index="277" value="0"/>
<variable name="BUILT(C079)" index="278" value="1"/>
<variable name="BUILT(C080)" index="279" value="1"/>
<variable name="BUILT(C081)" index="280" value="0"/>
<variable name="BUILT(C082)" index="281" value="0"/>
<variable name="BUILT(C083)" index="282" value="0"/>
<variable name="BUILT(C084)" index="283" value="0"/>
<variable name="BUILT(C085)" index="284" value="0"/>
<variable name="BUILT(C086)" index="285" value="0"/>
<variable name="BUILT(C087)" index="286" value="0"/>
<variable name="BUILT(C088)" index="287" value="1"/>
<variable name="BUILT(C089)" index="288" value="1"/>
<variable name="BUILT(C090)" index="289" value="0"/>
<variable name="BUILT(C091)" index="290" value="0"/>
<variable name="BUILT(C092)" index="291" value="0"/>
<variable name="BUILT(C093)" index="292" value="0"/>
<variable name="BUILT(C094)" index="293" value="0"/>
<variable name="BUILT(C095)" index="294" value="1"/>
<variable name="BUILT(C096)" index="295" value="0"/>
<variable name="BUILT(C097)" index="296" value="0"/>
<variable name="BUILT(C098)" index="297" value="0"/>
<variable name="BUILT(C099)" index="298" value="0"/>
<variable name="BUILT(C100)" index="299" value="0"/>
<variable name="BUILT(D001)" index="300" value="0"/>
<variable name="BUILT(D002)" index="301" value="0"/>
<variable name="BUILT(D003)" index="302" value="1"/>
<variable name="BUILT(D004)" index="303" value="0"/>
<variable name="BUILT(D005)" index="304" value="1"/>
<variable name="BUILT(D006)" index="305" value="0"/>
<variable name="BUILT(D007)" index="306" value="0"/>
<variable name="BUILT(D008)" index="307" value="0"/>
```

<variable name="BUILT(D009)" index="308" value="0"/>
<variable name="BUILT(D010)" index="309" value="0"/>
<variable name="BUILT(D011)" index="310" value="0"/>
<variable name="BUILT(D012)" index="311" value="0"/>
<variable name="BUILT(D013)" index="312" value="0"/>
<variable name="BUILT(D014)" index="313" value="0"/>
<variable name="BUILT(D015)" index="314" value="0"/>
<variable name="BUILT(D016)" index="315" value="0"/>
<variable name="BUILT(D017)" index="316" value="0"/>
<variable name="BUILT(D018)" index="317" value="0"/>
<variable name="BUILT(D019)" index="318" value="0"/>
<variable name="BUILT(D020)" index="319" value="1"/>
<variable name="BUILT(D021)" index="320" value="0"/>
<variable name="BUILT(D022)" index="321" value="0"/>
<variable name="BUILT(D023)" index="322" value="1"/>
<variable name="BUILT(D024)" index="323" value="0"/>
<variable name="BUILT(D025)" index="324" value="1"/>
<variable name="BUILT(D026)" index="325" value="1"/>
<variable name="BUILT(D027)" index="326" value="1"/>
<variable name="BUILT(D028)" index="327" value="1"/>
<variable name="BUILT(D029)" index="328" value="0"/>
<variable name="BUILT(D030)" index="329" value="1"/>
<variable name="BUILT(D031)" index="330" value="0"/>
<variable name="BUILT(D032)" index="331" value="0"/>
<variable name="BUILT(D033)" index="332" value="0"/>
<variable name="BUILT(D034)" index="333" value="0"/>
<variable name="BUILT(D035)" index="334" value="0"/>
<variable name="BUILT(D036)" index="335" value="0"/>
<variable name="BUILT(D037)" index="336" value="0"/>
<variable name="BUILT(D038)" index="337" value="0"/>
<variable name="BUILT(D039)" index="338" value="1"/>
<variable name="BUILT(D040)" index="339" value="0"/>
<variable name="BUILT(D041)" index="340" value="0"/>
<variable name="BUILT(D042)" index="341" value="0"/>
<variable name="BUILT(D043)" index="342" value="0"/>
<variable name="BUILT(D044)" index="343" value="0"/>
<variable name="BUILT(D045)" index="344" value="0"/>
<variable name="BUILT(D046)" index="345" value="1"/>
<variable name="BUILT(D047)" index="346" value="0"/>
<variable name="BUILT(D048)" index="347" value="0"/>
<variable name="BUILT(D049)" index="348" value="0"/>
<variable name="BUILT(D050)" index="349" value="1"/>
<variable name="BUILT(D051)" index="350" value="0"/>
<variable name="BUILT(D052)" index="351" value="0"/>
<variable name="BUILT(D053)" index="352" value="1"/>
<variable name="BUILT(D054)" index="353" value="0"/>
<variable name="BUILT(D055)" index="354" value="0"/>
<variable name="BUILT(D056)" index="355" value="1"/>
<variable name="BUILT(D057)" index="356" value="0"/>
<variable name="BUILT(D058)" index="357" value="0"/>
<variable name="BUILT(D059)" index="358" value="0"/>
<variable name="BUILT(D060)" index="359" value="1"/>
<variable name="BUILT(D061)" index="360" value="1"/>

<variable name="BUILT(D062)" index="361" value="0"/>
<variable name="BUILT(D063)" index="362" value="0"/>
<variable name="BUILT(D064)" index="363" value="0"/>
<variable name="BUILT(D065)" index="364" value="1"/>
<variable name="BUILT(D066)" index="365" value="0"/>
<variable name="BUILT(D067)" index="366" value="0"/>
<variable name="BUILT(D068)" index="367" value="0"/>
<variable name="BUILT(D069)" index="368" value="0"/>
<variable name="BUILT(D070)" index="369" value="0"/>
<variable name="BUILT(D071)" index="370" value="0"/>
<variable name="BUILT(D072)" index="371" value="0"/>
<variable name="BUILT(D073)" index="372" value="0"/>
<variable name="BUILT(D074)" index="373" value="0"/>
<variable name="BUILT(D075)" index="374" value="0"/>
<variable name="BUILT(D076)" index="375" value="0"/>
<variable name="BUILT(D077)" index="376" value="0"/>
<variable name="BUILT(D078)" index="377" value="0"/>
<variable name="BUILT(D079)" index="378" value="0"/>
<variable name="BUILT(D080)" index="379" value="0"/>
<variable name="BUILT(D081)" index="380" value="0"/>
<variable name="BUILT(D082)" index="381" value="0"/>
<variable name="BUILT(D083)" index="382" value="0"/>
<variable name="BUILT(D084)" index="383" value="0"/>
<variable name="BUILT(D085)" index="384" value="0"/>
<variable name="BUILT(D086)" index="385" value="0"/>
<variable name="BUILT(D087)" index="386" value="1"/>
<variable name="BUILT(D088)" index="387" value="0"/>
<variable name="BUILT(D089)" index="388" value="1"/>
<variable name="BUILT(D090)" index="389" value="0"/>
<variable name="BUILT(D091)" index="390" value="0"/>
<variable name="BUILT(D092)" index="391" value="0"/>
<variable name="BUILT(D093)" index="392" value="0"/>
<variable name="BUILT(D094)" index="393" value="1"/>
<variable name="BUILT(D095)" index="394" value="0"/>
<variable name="BUILT(D096)" index="395" value="0"/>
<variable name="BUILT(D097)" index="396" value="0"/>
<variable name="BUILT(D098)" index="397" value="0"/>
<variable name="BUILT(D099)" index="398" value="0"/>
<variable name="BUILT(D100)" index="399" value="1"/>
<variable name="BUILT(E001)" index="400" value="0"/>
<variable name="BUILT(E002)" index="401" value="0"/>
<variable name="BUILT(E003)" index="402" value="1"/>
<variable name="BUILT(E004)" index="403" value="0"/>
<variable name="BUILT(E005)" index="404" value="0"/>
<variable name="BUILT(E006)" index="405" value="0"/>
<variable name="BUILT(E007)" index="406" value="0"/>
<variable name="BUILT(E008)" index="407" value="0"/>
<variable name="BUILT(E009)" index="408" value="0"/>
<variable name="BUILT(E010)" index="409" value="0"/>
<variable name="BUILT(E011)" index="410" value="0"/>
<variable name="BUILT(E012)" index="411" value="0"/>
<variable name="BUILT(E013)" index="412" value="0"/>
<variable name="BUILT(E014)" index="413" value="0"/>

<variable name="BUILT(E015)" index="414" value="1"/>
<variable name="BUILT(E016)" index="415" value="0"/>
<variable name="BUILT(E017)" index="416" value="0"/>
<variable name="BUILT(E018)" index="417" value="1"/>
<variable name="BUILT(E019)" index="418" value="0"/>
<variable name="BUILT(E020)" index="419" value="0"/>
<variable name="BUILT(E021)" index="420" value="0"/>
<variable name="BUILT(E022)" index="421" value="0"/>
<variable name="BUILT(E023)" index="422" value="0"/>
<variable name="BUILT(E024)" index="423" value="1"/>
<variable name="BUILT(E025)" index="424" value="1"/>
<variable name="BUILT(E026)" index="425" value="1"/>
<variable name="BUILT(E027)" index="426" value="0"/>
<variable name="BUILT(E028)" index="427" value="0"/>
<variable name="BUILT(E029)" index="428" value="0"/>
<variable name="BUILT(E030)" index="429" value="1"/>
<variable name="BUILT(E031)" index="430" value="0"/>
<variable name="BUILT(E032)" index="431" value="0"/>
<variable name="BUILT(E033)" index="432" value="0"/>
<variable name="BUILT(E034)" index="433" value="0"/>
<variable name="BUILT(E035)" index="434" value="0"/>
<variable name="BUILT(E036)" index="435" value="0"/>
<variable name="BUILT(E037)" index="436" value="0"/>
<variable name="BUILT(E038)" index="437" value="0"/>
<variable name="BUILT(E039)" index="438" value="0"/>
<variable name="BUILT(E040)" index="439" value="0"/>
<variable name="BUILT(E041)" index="440" value="0"/>
<variable name="BUILT(E042)" index="441" value="0"/>
<variable name="BUILT(E043)" index="442" value="0"/>
<variable name="BUILT(E044)" index="443" value="0"/>
<variable name="BUILT(E045)" index="444" value="0"/>
<variable name="BUILT(E046)" index="445" value="0"/>
<variable name="BUILT(E047)" index="446" value="0"/>
<variable name="BUILT(E048)" index="447" value="0"/>
<variable name="BUILT(E049)" index="448" value="1"/>
<variable name="BUILT(E050)" index="449" value="0"/>
<variable name="BUILT(E051)" index="450" value="0"/>
<variable name="BUILT(E052)" index="451" value="0"/>
<variable name="BUILT(E053)" index="452" value="0"/>
<variable name="BUILT(E054)" index="453" value="1"/>
<variable name="BUILT(E055)" index="454" value="1"/>
<variable name="BUILT(E056)" index="455" value="0"/>
<variable name="BUILT(E057)" index="456" value="0"/>
<variable name="BUILT(E058)" index="457" value="0"/>
<variable name="BUILT(E059)" index="458" value="0"/>
<variable name="BUILT(E060)" index="459" value="0"/>
<variable name="BUILT(E061)" index="460" value="0"/>
<variable name="BUILT(E062)" index="461" value="0"/>
<variable name="BUILT(E063)" index="462" value="0"/>
<variable name="BUILT(E064)" index="463" value="0"/>
<variable name="BUILT(E065)" index="464" value="0"/>
<variable name="BUILT(E066)" index="465" value="1"/>
<variable name="BUILT(E067)" index="466" value="0"/>

```
<variable name="BUILT(E068)" index="467" value="0"/>
<variable name="BUILT(E069)" index="468" value="0"/>
<variable name="BUILT(E070)" index="469" value="0"/>
<variable name="BUILT(E071)" index="470" value="1"/>
<variable name="BUILT(E072)" index="471" value="0"/>
<variable name="BUILT(E073)" index="472" value="1"/>
<variable name="BUILT(E074)" index="473" value="0"/>
<variable name="BUILT(E075)" index="474" value="0"/>
<variable name="BUILT(E076)" index="475" value="0"/>
<variable name="BUILT(E077)" index="476" value="0"/>
<variable name="BUILT(E078)" index="477" value="1"/>
<variable name="BUILT(E079)" index="478" value="0"/>
<variable name="BUILT(E080)" index="479" value="0"/>
<variable name="BUILT(E081)" index="480" value="0"/>
<variable name="BUILT(E082)" index="481" value="1"/>
<variable name="BUILT(E083)" index="482" value="1"/>
<variable name="BUILT(E084)" index="483" value="0"/>
<variable name="BUILT(E085)" index="484" value="1"/>
<variable name="BUILT(E086)" index="485" value="0"/>
<variable name="BUILT(E087)" index="486" value="0"/>
<variable name="BUILT(E088)" index="487" value="0"/>
<variable name="BUILT(E089)" index="488" value="1"/>
<variable name="BUILT(E090)" index="489" value="0"/>
<variable name="BUILT(E091)" index="490" value="0"/>
<variable name="BUILT(E092)" index="491" value="0"/>
<variable name="BUILT(E093)" index="492" value="0"/>
<variable name="BUILT(E094)" index="493" value="0"/>
<variable name="BUILT(E095)" index="494" value="1"/>
<variable name="BUILT(E096)" index="495" value="0"/>
<variable name="BUILT(E097)" index="496" value="0"/>
<variable name="BUILT(E098)" index="497" value="1"/>
<variable name="BUILT(E099)" index="498" value="1"/>
<variable name="BUILT(E100)" index="499" value="0"/>
```

3.2 Output for Assignment-handling Variables

```
<variable name="ASSIGN(A001,A001)" index="500" value="0"/>
<variable name="ASSIGN(A001,A002)" index="501" value="0"/>
<variable name="ASSIGN(A001,A003)" index="502" value="0"/>
<variable name="ASSIGN(A001,A004)" index="503" value="0"/>
<variable name="ASSIGN(A001,A005)" index="504" value="0"/>
<variable name="ASSIGN(A001,A006)" index="505" value="0"/>
<variable name="ASSIGN(A001,A007)" index="506" value="0"/>
<variable name="ASSIGN(A001,A008)" index="507" value="0"/>
<variable name="ASSIGN(A001,A009)" index="508" value="0"/>
<variable name="ASSIGN(A001,A010)" index="509" value="0"/>
<variable name="ASSIGN(A001,A011)" index="510" value="0"/>
<variable name="ASSIGN(A001,A012)" index="511" value="0"/>
<variable name="ASSIGN(A001,A013)" index="512" value="0"/>
<variable name="ASSIGN(A001,A014)" index="513" value="0"/>
<variable name="ASSIGN(A001,A015)" index="514" value="0"/>
<variable name="ASSIGN(A001,A016)" index="515" value="0"/>
<variable name="ASSIGN(A001,A017)" index="516" value="0"/>
<variable name="ASSIGN(A001,A018)" index="517" value="0"/>
<variable name="ASSIGN(A001,A019)" index="518" value="0"/>
<variable name="ASSIGN(A001,A020)" index="519" value="0"/>
<variable name="ASSIGN(A001,A021)" index="520" value="0"/>
```

<variable name="ASSIGN(A001,A022)" index="521" value="0"/>
<variable name="ASSIGN(A001,A023)" index="522" value="0"/>
<variable name="ASSIGN(A001,A024)" index="523" value="0"/>
<variable name="ASSIGN(A001,A025)" index="524" value="0"/>
<variable name="ASSIGN(A001,A026)" index="525" value="0"/>
<variable name="ASSIGN(A001,A027)" index="526" value="0"/>
<variable name="ASSIGN(A001,A028)" index="527" value="0"/>
<variable name="ASSIGN(A001,A029)" index="528" value="0"/>
<variable name="ASSIGN(A001,A030)" index="529" value="0"/>
<variable name="ASSIGN(A001,A031)" index="530" value="0"/>
<variable name="ASSIGN(A001,A032)" index="531" value="0"/>
<variable name="ASSIGN(A001,A033)" index="532" value="0"/>
<variable name="ASSIGN(A001,A034)" index="533" value="0"/>
<variable name="ASSIGN(A001,A035)" index="534" value="0"/>
<variable name="ASSIGN(A001,A036)" index="535" value="0"/>
<variable name="ASSIGN(A001,A037)" index="536" value="0"/>
<variable name="ASSIGN(A001,A038)" index="537" value="0"/>
<variable name="ASSIGN(A001,A039)" index="538" value="0"/>
<variable name="ASSIGN(A001,A040)" index="539" value="0"/>
<variable name="ASSIGN(A001,A041)" index="540" value="0"/>
<variable name="ASSIGN(A001,A042)" index="541" value="0"/>
<variable name="ASSIGN(A001,A043)" index="542" value="0"/>
<variable name="ASSIGN(A001,A044)" index="543" value="0"/>
<variable name="ASSIGN(A001,A045)" index="544" value="0"/>
<variable name="ASSIGN(A001,A046)" index="545" value="0"/>
<variable name="ASSIGN(A001,A047)" index="546" value="0"/>
<variable name="ASSIGN(A001,A048)" index="547" value="0"/>
<variable name="ASSIGN(A001,A049)" index="548" value="0"/>
<variable name="ASSIGN(A001,A050)" index="549" value="0"/>
<variable name="ASSIGN(A001,A051)" index="550" value="0"/>
<variable name="ASSIGN(A001,A052)" index="551" value="0"/>
<variable name="ASSIGN(A001,A053)" index="552" value="0"/>
<variable name="ASSIGN(A001,A054)" index="553" value="0"/>
<variable name="ASSIGN(A001,A055)" index="554" value="0"/>
<variable name="ASSIGN(A001,A056)" index="555" value="0"/>
<variable name="ASSIGN(A001,A057)" index="556" value="0"/>
<variable name="ASSIGN(A001,A058)" index="557" value="0"/>
<variable name="ASSIGN(A001,A059)" index="558" value="0"/>
<variable name="ASSIGN(A001,A060)" index="559" value="0"/>
<variable name="ASSIGN(A001,A061)" index="560" value="0"/>
<variable name="ASSIGN(A001,A062)" index="561" value="0"/>
<variable name="ASSIGN(A001,A063)" index="562" value="0"/>
<variable name="ASSIGN(A001,A064)" index="563" value="0"/>
<variable name="ASSIGN(A001,A065)" index="564" value="0"/>
<variable name="ASSIGN(A001,A066)" index="565" value="0"/>
<variable name="ASSIGN(A001,A067)" index="566" value="0"/>
<variable name="ASSIGN(A001,A068)" index="567" value="0"/>
<variable name="ASSIGN(A001,A069)" index="568" value="0"/>
<variable name="ASSIGN(A001,A070)" index="569" value="1"/>
<variable name="ASSIGN(A001,A071)" index="570" value="0"/>
<variable name="ASSIGN(A001,A072)" index="571" value="0"/>
<variable name="ASSIGN(A001,A073)" index="572" value="0"/>
<variable name="ASSIGN(A001,A074)" index="573" value="0"/>
<variable name="ASSIGN(A001,A075)" index="574" value="0"/>
<variable name="ASSIGN(A001,A076)" index="575" value="0"/>
<variable name="ASSIGN(A001,A077)" index="576" value="0"/>
<variable name="ASSIGN(A001,A078)" index="577" value="0"/>
<variable name="ASSIGN(A001,A079)" index="578" value="0"/>
<variable name="ASSIGN(A001,A080)" index="579" value="0"/>
<variable name="ASSIGN(A001,A081)" index="580" value="0"/>
<variable name="ASSIGN(A001,A082)" index="581" value="0"/>
<variable name="ASSIGN(A001,A083)" index="582" value="0"/>
<variable name="ASSIGN(A001,A084)" index="583" value="0"/>
<variable name="ASSIGN(A001,A085)" index="584" value="0"/>
<variable name="ASSIGN(A001,A086)" index="585" value="0"/>
<variable name="ASSIGN(A001,A087)" index="586" value="0"/>
<variable name="ASSIGN(A001,A088)" index="587" value="0"/>
<variable name="ASSIGN(A001,A089)" index="588" value="0"/>
<variable name="ASSIGN(A001,A090)" index="589" value="0"/>
<variable name="ASSIGN(A001,A091)" index="590" value="0"/>
<variable name="ASSIGN(A001,A092)" index="591" value="0"/>
<variable name="ASSIGN(A001,A093)" index="592" value="0"/>

<variable name="ASSIGN(A001,A094)" index="593" value="0"/>
<variable name="ASSIGN(A001,A095)" index="594" value="0"/>
<variable name="ASSIGN(A001,A096)" index="595" value="0"/>
<variable name="ASSIGN(A001,A097)" index="596" value="0"/>
<variable name="ASSIGN(A001,A098)" index="597" value="0"/>
<variable name="ASSIGN(A001,A099)" index="598" value="0"/>
<variable name="ASSIGN(A001,A100)" index="599" value="0"/>
<variable name="ASSIGN(A001,B001)" index="600" value="0"/>
<variable name="ASSIGN(A001,B002)" index="601" value="0"/>
<variable name="ASSIGN(A001,B003)" index="602" value="0"/>
<variable name="ASSIGN(A001,B004)" index="603" value="0"/>
<variable name="ASSIGN(A001,B005)" index="604" value="0"/>
<variable name="ASSIGN(A001,B006)" index="605" value="0"/>
<variable name="ASSIGN(A001,B007)" index="606" value="0"/>
<variable name="ASSIGN(A001,B008)" index="607" value="0"/>
<variable name="ASSIGN(A001,B009)" index="608" value="0"/>
<variable name="ASSIGN(A001,B010)" index="609" value="0"/>
<variable name="ASSIGN(A001,B011)" index="610" value="0"/>
<variable name="ASSIGN(A001,B012)" index="611" value="0"/>
<variable name="ASSIGN(A001,B013)" index="612" value="0"/>
<variable name="ASSIGN(A001,B014)" index="613" value="0"/>
<variable name="ASSIGN(A001,B015)" index="614" value="0"/>
<variable name="ASSIGN(A001,B016)" index="615" value="0"/>
<variable name="ASSIGN(A001,B017)" index="616" value="0"/>
<variable name="ASSIGN(A001,B018)" index="617" value="0"/>
<variable name="ASSIGN(A001,B019)" index="618" value="0"/>
<variable name="ASSIGN(A001,B020)" index="619" value="0"/>
<variable name="ASSIGN(A001,B021)" index="620" value="0"/>
<variable name="ASSIGN(A001,B022)" index="621" value="0"/>
<variable name="ASSIGN(A001,B023)" index="622" value="0"/>
<variable name="ASSIGN(A001,B024)" index="623" value="0"/>
<variable name="ASSIGN(A001,B025)" index="624" value="0"/>
<variable name="ASSIGN(A001,B026)" index="625" value="0"/>
<variable name="ASSIGN(A001,B027)" index="626" value="0"/>
<variable name="ASSIGN(A001,B028)" index="627" value="0"/>
<variable name="ASSIGN(A001,B029)" index="628" value="0"/>
<variable name="ASSIGN(A001,B030)" index="629" value="0"/>
<variable name="ASSIGN(A001,B031)" index="630" value="0"/>
<variable name="ASSIGN(A001,B032)" index="631" value="0"/>
<variable name="ASSIGN(A001,B033)" index="632" value="0"/>
<variable name="ASSIGN(A001,B034)" index="633" value="0"/>
<variable name="ASSIGN(A001,B035)" index="634" value="0"/>
<variable name="ASSIGN(A001,B036)" index="635" value="0"/>
<variable name="ASSIGN(A001,B037)" index="636" value="0"/>
<variable name="ASSIGN(A001,B038)" index="637" value="0"/>
<variable name="ASSIGN(A001,B039)" index="638" value="0"/>
<variable name="ASSIGN(A001,B040)" index="639" value="0"/>
<variable name="ASSIGN(A001,B041)" index="640" value="0"/>
<variable name="ASSIGN(A001,B042)" index="641" value="0"/>
<variable name="ASSIGN(A001,B043)" index="642" value="0"/>
<variable name="ASSIGN(A001,B044)" index="643" value="0"/>
<variable name="ASSIGN(A001,B045)" index="644" value="0"/>
<variable name="ASSIGN(A001,B046)" index="645" value="0"/>
<variable name="ASSIGN(A001,B047)" index="646" value="0"/>
<variable name="ASSIGN(A001,B048)" index="647" value="0"/>
<variable name="ASSIGN(A001,B049)" index="648" value="0"/>
<variable name="ASSIGN(A001,B050)" index="649" value="0"/>
<variable name="ASSIGN(A001,B051)" index="650" value="0"/>
<variable name="ASSIGN(A001,B052)" index="651" value="0"/>
<variable name="ASSIGN(A001,B053)" index="652" value="0"/>
<variable name="ASSIGN(A001,B054)" index="653" value="0"/>
<variable name="ASSIGN(A001,B055)" index="654" value="0"/>
<variable name="ASSIGN(A001,B056)" index="655" value="0"/>
<variable name="ASSIGN(A001,B057)" index="656" value="0"/>
<variable name="ASSIGN(A001,B058)" index="657" value="0"/>
<variable name="ASSIGN(A001,B059)" index="658" value="0"/>
<variable name="ASSIGN(A001,B060)" index="659" value="0"/>
<variable name="ASSIGN(A001,B061)" index="660" value="0"/>
<variable name="ASSIGN(A001,B062)" index="661" value="0"/>
<variable name="ASSIGN(A001,B063)" index="662" value="0"/>
<variable name="ASSIGN(A001,B064)" index="663" value="0"/>
<variable name="ASSIGN(A001,B065)" index="664" value="0"/>

<variable name="ASSIGN(A001,B066)" index="665" value="0"/>
<variable name="ASSIGN(A001,B067)" index="666" value="0"/>
<variable name="ASSIGN(A001,B068)" index="667" value="0"/>
<variable name="ASSIGN(A001,B069)" index="668" value="0"/>
<variable name="ASSIGN(A001,B070)" index="669" value="0"/>
<variable name="ASSIGN(A001,B071)" index="670" value="0"/>
<variable name="ASSIGN(A001,B072)" index="671" value="0"/>
<variable name="ASSIGN(A001,B073)" index="672" value="0"/>
<variable name="ASSIGN(A001,B074)" index="673" value="0"/>
<variable name="ASSIGN(A001,B075)" index="674" value="0"/>
<variable name="ASSIGN(A001,B076)" index="675" value="0"/>
<variable name="ASSIGN(A001,B077)" index="676" value="0"/>
<variable name="ASSIGN(A001,B078)" index="677" value="0"/>
<variable name="ASSIGN(A001,B079)" index="678" value="0"/>
<variable name="ASSIGN(A001,B080)" index="679" value="0"/>
<variable name="ASSIGN(A001,B081)" index="680" value="0"/>
<variable name="ASSIGN(A001,B082)" index="681" value="0"/>
<variable name="ASSIGN(A001,B083)" index="682" value="0"/>
<variable name="ASSIGN(A001,B084)" index="683" value="0"/>
<variable name="ASSIGN(A001,B085)" index="684" value="0"/>
<variable name="ASSIGN(A001,B086)" index="685" value="0"/>
<variable name="ASSIGN(A001,B087)" index="686" value="0"/>
<variable name="ASSIGN(A001,B088)" index="687" value="0"/>
<variable name="ASSIGN(A001,B089)" index="688" value="0"/>
<variable name="ASSIGN(A001,B090)" index="689" value="0"/>
<variable name="ASSIGN(A001,B091)" index="690" value="0"/>
<variable name="ASSIGN(A001,B092)" index="691" value="0"/>
<variable name="ASSIGN(A001,B093)" index="692" value="0"/>
<variable name="ASSIGN(A001,B094)" index="693" value="0"/>
<variable name="ASSIGN(A001,B095)" index="694" value="0"/>
<variable name="ASSIGN(A001,B096)" index="695" value="0"/>
<variable name="ASSIGN(A001,B097)" index="696" value="0"/>
<variable name="ASSIGN(A001,B098)" index="697" value="0"/>
<variable name="ASSIGN(A001,B099)" index="698" value="0"/>
<variable name="ASSIGN(A001,B100)" index="699" value="0"/>
<variable name="ASSIGN(A001,C001)" index="700" value="0"/>
<variable name="ASSIGN(A001,C002)" index="701" value="0"/>
<variable name="ASSIGN(A001,C003)" index="702" value="0"/>
<variable name="ASSIGN(A001,C004)" index="703" value="0"/>
<variable name="ASSIGN(A001,C005)" index="704" value="0"/>
<variable name="ASSIGN(A001,C006)" index="705" value="0"/>
<variable name="ASSIGN(A001,C007)" index="706" value="0"/>
<variable name="ASSIGN(A001,C008)" index="707" value="0"/>
<variable name="ASSIGN(A001,C009)" index="708" value="0"/>
<variable name="ASSIGN(A001,C010)" index="709" value="0"/>
<variable name="ASSIGN(A001,C011)" index="710" value="0"/>
<variable name="ASSIGN(A001,C012)" index="711" value="0"/>
<variable name="ASSIGN(A001,C013)" index="712" value="0"/>
<variable name="ASSIGN(A001,C014)" index="713" value="0"/>
<variable name="ASSIGN(A001,C015)" index="714" value="0"/>
<variable name="ASSIGN(A001,C016)" index="715" value="0"/>
<variable name="ASSIGN(A001,C017)" index="716" value="0"/>
<variable name="ASSIGN(A001,C018)" index="717" value="0"/>
<variable name="ASSIGN(A001,C019)" index="718" value="0"/>
<variable name="ASSIGN(A001,C020)" index="719" value="0"/>
<variable name="ASSIGN(A001,C021)" index="720" value="0"/>
<variable name="ASSIGN(A001,C022)" index="721" value="0"/>
<variable name="ASSIGN(A001,C023)" index="722" value="0"/>
<variable name="ASSIGN(A001,C024)" index="723" value="0"/>
<variable name="ASSIGN(A001,C025)" index="724" value="0"/>
<variable name="ASSIGN(A001,C026)" index="725" value="0"/>
<variable name="ASSIGN(A001,C027)" index="726" value="0"/>
<variable name="ASSIGN(A001,C028)" index="727" value="0"/>
<variable name="ASSIGN(A001,C029)" index="728" value="0"/>
<variable name="ASSIGN(A001,C030)" index="729" value="0"/>
<variable name="ASSIGN(A001,C031)" index="730" value="0"/>
<variable name="ASSIGN(A001,C032)" index="731" value="0"/>
<variable name="ASSIGN(A001,C033)" index="732" value="0"/>
<variable name="ASSIGN(A001,C034)" index="733" value="0"/>
<variable name="ASSIGN(A001,C035)" index="734" value="0"/>
<variable name="ASSIGN(A001,C036)" index="735" value="0"/>
<variable name="ASSIGN(A001,C037)" index="736" value="0"/>

<variable name="ASSIGN(A001,C038)" index="737" value="0"/>
<variable name="ASSIGN(A001,C039)" index="738" value="0"/>
<variable name="ASSIGN(A001,C040)" index="739" value="0"/>
<variable name="ASSIGN(A001,C041)" index="740" value="0"/>
<variable name="ASSIGN(A001,C042)" index="741" value="0"/>
<variable name="ASSIGN(A001,C043)" index="742" value="0"/>
<variable name="ASSIGN(A001,C044)" index="743" value="0"/>
<variable name="ASSIGN(A001,C045)" index="744" value="0"/>
<variable name="ASSIGN(A001,C046)" index="745" value="0"/>
<variable name="ASSIGN(A001,C047)" index="746" value="0"/>
<variable name="ASSIGN(A001,C048)" index="747" value="0"/>
<variable name="ASSIGN(A001,C049)" index="748" value="0"/>
<variable name="ASSIGN(A001,C050)" index="749" value="0"/>
<variable name="ASSIGN(A001,C051)" index="750" value="0"/>
<variable name="ASSIGN(A001,C052)" index="751" value="0"/>
<variable name="ASSIGN(A001,C053)" index="752" value="0"/>
<variable name="ASSIGN(A001,C054)" index="753" value="0"/>
<variable name="ASSIGN(A001,C055)" index="754" value="0"/>
<variable name="ASSIGN(A001,C056)" index="755" value="0"/>
<variable name="ASSIGN(A001,C057)" index="756" value="0"/>
<variable name="ASSIGN(A001,C058)" index="757" value="0"/>
<variable name="ASSIGN(A001,C059)" index="758" value="0"/>
<variable name="ASSIGN(A001,C060)" index="759" value="0"/>
<variable name="ASSIGN(A001,C061)" index="760" value="0"/>
<variable name="ASSIGN(A001,C062)" index="761" value="0"/>
<variable name="ASSIGN(A001,C063)" index="762" value="0"/>
<variable name="ASSIGN(A001,C064)" index="763" value="0"/>
<variable name="ASSIGN(A001,C065)" index="764" value="0"/>
<variable name="ASSIGN(A001,C066)" index="765" value="0"/>
<variable name="ASSIGN(A001,C067)" index="766" value="0"/>
<variable name="ASSIGN(A001,C068)" index="767" value="0"/>
<variable name="ASSIGN(A001,C069)" index="768" value="0"/>
<variable name="ASSIGN(A001,C070)" index="769" value="0"/>
<variable name="ASSIGN(A001,C071)" index="770" value="0"/>
<variable name="ASSIGN(A001,C072)" index="771" value="0"/>
<variable name="ASSIGN(A001,C073)" index="772" value="0"/>
<variable name="ASSIGN(A001,C074)" index="773" value="0"/>
<variable name="ASSIGN(A001,C075)" index="774" value="0"/>
<variable name="ASSIGN(A001,C076)" index="775" value="0"/>
<variable name="ASSIGN(A001,C077)" index="776" value="0"/>
<variable name="ASSIGN(A001,C078)" index="777" value="0"/>
<variable name="ASSIGN(A001,C079)" index="778" value="0"/>
<variable name="ASSIGN(A001,C080)" index="779" value="0"/>
<variable name="ASSIGN(A001,C081)" index="780" value="0"/>
<variable name="ASSIGN(A001,C082)" index="781" value="0"/>
<variable name="ASSIGN(A001,C083)" index="782" value="0"/>
<variable name="ASSIGN(A001,C084)" index="783" value="0"/>
<variable name="ASSIGN(A001,C085)" index="784" value="0"/>
<variable name="ASSIGN(A001,C086)" index="785" value="0"/>
<variable name="ASSIGN(A001,C087)" index="786" value="0"/>
<variable name="ASSIGN(A001,C088)" index="787" value="0"/>
<variable name="ASSIGN(A001,C089)" index="788" value="0"/>
<variable name="ASSIGN(A001,C090)" index="789" value="0"/>
<variable name="ASSIGN(A001,C091)" index="790" value="0"/>
<variable name="ASSIGN(A001,C092)" index="791" value="0"/>
<variable name="ASSIGN(A001,C093)" index="792" value="0"/>
<variable name="ASSIGN(A001,C094)" index="793" value="0"/>
<variable name="ASSIGN(A001,C095)" index="794" value="0"/>
<variable name="ASSIGN(A001,C096)" index="795" value="0"/>
<variable name="ASSIGN(A001,C097)" index="796" value="0"/>
<variable name="ASSIGN(A001,C098)" index="797" value="0"/>
<variable name="ASSIGN(A001,C099)" index="798" value="0"/>
<variable name="ASSIGN(A001,C100)" index="799" value="0"/>
<variable name="ASSIGN(A001,D001)" index="800" value="0"/>
<variable name="ASSIGN(A001,D002)" index="801" value="0"/>
<variable name="ASSIGN(A001,D003)" index="802" value="0"/>
<variable name="ASSIGN(A001,D004)" index="803" value="0"/>
<variable name="ASSIGN(A001,D005)" index="804" value="0"/>
<variable name="ASSIGN(A001,D006)" index="805" value="0"/>
<variable name="ASSIGN(A001,D007)" index="806" value="0"/>
<variable name="ASSIGN(A001,D008)" index="807" value="0"/>
<variable name="ASSIGN(A001,D009)" index="808" value="0"/>

<variable name="ASSIGN(A001,D010)" index="809" value="0"/>
<variable name="ASSIGN(A001,D011)" index="810" value="0"/>
<variable name="ASSIGN(A001,D012)" index="811" value="0"/>
<variable name="ASSIGN(A001,D013)" index="812" value="0"/>
<variable name="ASSIGN(A001,D014)" index="813" value="0"/>
<variable name="ASSIGN(A001,D015)" index="814" value="0"/>
<variable name="ASSIGN(A001,D016)" index="815" value="0"/>
<variable name="ASSIGN(A001,D017)" index="816" value="0"/>
<variable name="ASSIGN(A001,D018)" index="817" value="0"/>
<variable name="ASSIGN(A001,D019)" index="818" value="0"/>
<variable name="ASSIGN(A001,D020)" index="819" value="0"/>
<variable name="ASSIGN(A001,D021)" index="820" value="0"/>
<variable name="ASSIGN(A001,D022)" index="821" value="0"/>
<variable name="ASSIGN(A001,D023)" index="822" value="0"/>
<variable name="ASSIGN(A001,D024)" index="823" value="0"/>
<variable name="ASSIGN(A001,D025)" index="824" value="0"/>
<variable name="ASSIGN(A001,D026)" index="825" value="0"/>
<variable name="ASSIGN(A001,D027)" index="826" value="0"/>
<variable name="ASSIGN(A001,D028)" index="827" value="0"/>
<variable name="ASSIGN(A001,D029)" index="828" value="0"/>
<variable name="ASSIGN(A001,D030)" index="829" value="0"/>
<variable name="ASSIGN(A001,D031)" index="830" value="0"/>
<variable name="ASSIGN(A001,D032)" index="831" value="0"/>
<variable name="ASSIGN(A001,D033)" index="832" value="0"/>
<variable name="ASSIGN(A001,D034)" index="833" value="0"/>
<variable name="ASSIGN(A001,D035)" index="834" value="0"/>
<variable name="ASSIGN(A001,D036)" index="835" value="0"/>
<variable name="ASSIGN(A001,D037)" index="836" value="0"/>
<variable name="ASSIGN(A001,D038)" index="837" value="0"/>
<variable name="ASSIGN(A001,D039)" index="838" value="0"/>
<variable name="ASSIGN(A001,D040)" index="839" value="0"/>
<variable name="ASSIGN(A001,D041)" index="840" value="0"/>
<variable name="ASSIGN(A001,D042)" index="841" value="0"/>
<variable name="ASSIGN(A001,D043)" index="842" value="0"/>
<variable name="ASSIGN(A001,D044)" index="843" value="0"/>
<variable name="ASSIGN(A001,D045)" index="844" value="0"/>
<variable name="ASSIGN(A001,D046)" index="845" value="0"/>
<variable name="ASSIGN(A001,D047)" index="846" value="0"/>
<variable name="ASSIGN(A001,D048)" index="847" value="0"/>
<variable name="ASSIGN(A001,D049)" index="848" value="0"/>
<variable name="ASSIGN(A001,D050)" index="849" value="0"/>
<variable name="ASSIGN(A001,D051)" index="850" value="0"/>
<variable name="ASSIGN(A001,D052)" index="851" value="0"/>
<variable name="ASSIGN(A001,D053)" index="852" value="0"/>
<variable name="ASSIGN(A001,D054)" index="853" value="0"/>
<variable name="ASSIGN(A001,D055)" index="854" value="0"/>
<variable name="ASSIGN(A001,D056)" index="855" value="0"/>
<variable name="ASSIGN(A001,D057)" index="856" value="0"/>
<variable name="ASSIGN(A001,D058)" index="857" value="0"/>
<variable name="ASSIGN(A001,D059)" index="858" value="0"/>
<variable name="ASSIGN(A001,D060)" index="859" value="0"/>
<variable name="ASSIGN(A001,D061)" index="860" value="0"/>
<variable name="ASSIGN(A001,D062)" index="861" value="0"/>
<variable name="ASSIGN(A001,D063)" index="862" value="0"/>
<variable name="ASSIGN(A001,D064)" index="863" value="0"/>
<variable name="ASSIGN(A001,D065)" index="864" value="0"/>
<variable name="ASSIGN(A001,D066)" index="865" value="0"/>
<variable name="ASSIGN(A001,D067)" index="866" value="0"/>
<variable name="ASSIGN(A001,D068)" index="867" value="0"/>
<variable name="ASSIGN(A001,D069)" index="868" value="0"/>
<variable name="ASSIGN(A001,D070)" index="869" value="0"/>
<variable name="ASSIGN(A001,D071)" index="870" value="0"/>
<variable name="ASSIGN(A001,D072)" index="871" value="0"/>
<variable name="ASSIGN(A001,D073)" index="872" value="0"/>
<variable name="ASSIGN(A001,D074)" index="873" value="0"/>
<variable name="ASSIGN(A001,D075)" index="874" value="0"/>
<variable name="ASSIGN(A001,D076)" index="875" value="0"/>
<variable name="ASSIGN(A001,D077)" index="876" value="0"/>
<variable name="ASSIGN(A001,D078)" index="877" value="0"/>
<variable name="ASSIGN(A001,D079)" index="878" value="0"/>
<variable name="ASSIGN(A001,D080)" index="879" value="0"/>
<variable name="ASSIGN(A001,D081)" index="880" value="0"/>

<variable name="ASSIGN(A001,D082)" index="881" value="0"/>
<variable name="ASSIGN(A001,D083)" index="882" value="0"/>
<variable name="ASSIGN(A001,D084)" index="883" value="0"/>
<variable name="ASSIGN(A001,D085)" index="884" value="0"/>
<variable name="ASSIGN(A001,D086)" index="885" value="0"/>
<variable name="ASSIGN(A001,D087)" index="886" value="0"/>
<variable name="ASSIGN(A001,D088)" index="887" value="0"/>
<variable name="ASSIGN(A001,D089)" index="888" value="0"/>
<variable name="ASSIGN(A001,D090)" index="889" value="0"/>
<variable name="ASSIGN(A001,D091)" index="890" value="0"/>
<variable name="ASSIGN(A001,D092)" index="891" value="0"/>
<variable name="ASSIGN(A001,D093)" index="892" value="0"/>
<variable name="ASSIGN(A001,D094)" index="893" value="0"/>
<variable name="ASSIGN(A001,D095)" index="894" value="0"/>
<variable name="ASSIGN(A001,D096)" index="895" value="0"/>
<variable name="ASSIGN(A001,D097)" index="896" value="0"/>
<variable name="ASSIGN(A001,D098)" index="897" value="0"/>
<variable name="ASSIGN(A001,D099)" index="898" value="0"/>
<variable name="ASSIGN(A001,D100)" index="899" value="0"/>
<variable name="ASSIGN(A001,E001)" index="900" value="0"/>
<variable name="ASSIGN(A001,E002)" index="901" value="0"/>
<variable name="ASSIGN(A001,E003)" index="902" value="0"/>
<variable name="ASSIGN(A001,E004)" index="903" value="0"/>
<variable name="ASSIGN(A001,E005)" index="904" value="0"/>
<variable name="ASSIGN(A001,E006)" index="905" value="0"/>
<variable name="ASSIGN(A001,E007)" index="906" value="0"/>
<variable name="ASSIGN(A001,E008)" index="907" value="0"/>
<variable name="ASSIGN(A001,E009)" index="908" value="0"/>
<variable name="ASSIGN(A001,E010)" index="909" value="0"/>
<variable name="ASSIGN(A001,E011)" index="910" value="0"/>
<variable name="ASSIGN(A001,E012)" index="911" value="0"/>
<variable name="ASSIGN(A001,E013)" index="912" value="0"/>
<variable name="ASSIGN(A001,E014)" index="913" value="0"/>
<variable name="ASSIGN(A001,E015)" index="914" value="0"/>
<variable name="ASSIGN(A001,E016)" index="915" value="0"/>
<variable name="ASSIGN(A001,E017)" index="916" value="0"/>
<variable name="ASSIGN(A001,E018)" index="917" value="0"/>
<variable name="ASSIGN(A001,E019)" index="918" value="0"/>
<variable name="ASSIGN(A001,E020)" index="919" value="0"/>
<variable name="ASSIGN(A001,E021)" index="920" value="0"/>
<variable name="ASSIGN(A001,E022)" index="921" value="0"/>
<variable name="ASSIGN(A001,E023)" index="922" value="0"/>
<variable name="ASSIGN(A001,E024)" index="923" value="0"/>
<variable name="ASSIGN(A001,E025)" index="924" value="0"/>
<variable name="ASSIGN(A001,E026)" index="925" value="0"/>
<variable name="ASSIGN(A001,E027)" index="926" value="0"/>
<variable name="ASSIGN(A001,E028)" index="927" value="0"/>
<variable name="ASSIGN(A001,E029)" index="928" value="0"/>
<variable name="ASSIGN(A001,E030)" index="929" value="0"/>
<variable name="ASSIGN(A001,E031)" index="930" value="0"/>
<variable name="ASSIGN(A001,E032)" index="931" value="0"/>
<variable name="ASSIGN(A001,E033)" index="932" value="0"/>
<variable name="ASSIGN(A001,E034)" index="933" value="0"/>
<variable name="ASSIGN(A001,E035)" index="934" value="0"/>
<variable name="ASSIGN(A001,E036)" index="935" value="0"/>
<variable name="ASSIGN(A001,E037)" index="936" value="0"/>
<variable name="ASSIGN(A001,E038)" index="937" value="0"/>
<variable name="ASSIGN(A001,E039)" index="938" value="0"/>
<variable name="ASSIGN(A001,E040)" index="939" value="0"/>
<variable name="ASSIGN(A001,E041)" index="940" value="0"/>
<variable name="ASSIGN(A001,E042)" index="941" value="0"/>
<variable name="ASSIGN(A001,E043)" index="942" value="0"/>
<variable name="ASSIGN(A001,E044)" index="943" value="0"/>
<variable name="ASSIGN(A001,E045)" index="944" value="0"/>
<variable name="ASSIGN(A001,E046)" index="945" value="0"/>
<variable name="ASSIGN(A001,E047)" index="946" value="0"/>
<variable name="ASSIGN(A001,E048)" index="947" value="0"/>
<variable name="ASSIGN(A001,E049)" index="948" value="0"/>
<variable name="ASSIGN(A001,E050)" index="949" value="0"/>
<variable name="ASSIGN(A001,E051)" index="950" value="0"/>
<variable name="ASSIGN(A001,E052)" index="951" value="0"/>
<variable name="ASSIGN(A001,E053)" index="952" value="0"/>

<variable name="ASSIGN(A001,E054)" index="953" value="0"/>
<variable name="ASSIGN(A001,E055)" index="954" value="0"/>
<variable name="ASSIGN(A001,E056)" index="955" value="0"/>
<variable name="ASSIGN(A001,E057)" index="956" value="0"/>
<variable name="ASSIGN(A001,E058)" index="957" value="0"/>
<variable name="ASSIGN(A001,E059)" index="958" value="0"/>
<variable name="ASSIGN(A001,E060)" index="959" value="0"/>
<variable name="ASSIGN(A001,E061)" index="960" value="0"/>
<variable name="ASSIGN(A001,E062)" index="961" value="0"/>
<variable name="ASSIGN(A001,E063)" index="962" value="0"/>
<variable name="ASSIGN(A001,E064)" index="963" value="0"/>
<variable name="ASSIGN(A001,E065)" index="964" value="0"/>
<variable name="ASSIGN(A001,E066)" index="965" value="0"/>
<variable name="ASSIGN(A001,E067)" index="966" value="0"/>
<variable name="ASSIGN(A001,E068)" index="967" value="0"/>
<variable name="ASSIGN(A001,E069)" index="968" value="0"/>
<variable name="ASSIGN(A001,E070)" index="969" value="0"/>
<variable name="ASSIGN(A001,E071)" index="970" value="0"/>
<variable name="ASSIGN(A001,E072)" index="971" value="0"/>
<variable name="ASSIGN(A001,E073)" index="972" value="0"/>
<variable name="ASSIGN(A001,E074)" index="973" value="0"/>
<variable name="ASSIGN(A001,E075)" index="974" value="0"/>
<variable name="ASSIGN(A001,E076)" index="975" value="0"/>
<variable name="ASSIGN(A001,E077)" index="976" value="0"/>
<variable name="ASSIGN(A001,E078)" index="977" value="0"/>
<variable name="ASSIGN(A001,E079)" index="978" value="0"/>
<variable name="ASSIGN(A001,E080)" index="979" value="0"/>
<variable name="ASSIGN(A001,E081)" index="980" value="0"/>
<variable name="ASSIGN(A001,E082)" index="981" value="0"/>
<variable name="ASSIGN(A001,E083)" index="982" value="0"/>
<variable name="ASSIGN(A001,E084)" index="983" value="0"/>
<variable name="ASSIGN(A001,E085)" index="984" value="0"/>
<variable name="ASSIGN(A001,E086)" index="985" value="0"/>
<variable name="ASSIGN(A001,E087)" index="986" value="0"/>
<variable name="ASSIGN(A001,E088)" index="987" value="0"/>
<variable name="ASSIGN(A001,E089)" index="988" value="0"/>
<variable name="ASSIGN(A001,E090)" index="989" value="0"/>
<variable name="ASSIGN(A001,E091)" index="990" value="0"/>
<variable name="ASSIGN(A001,E092)" index="991" value="0"/>
<variable name="ASSIGN(A001,E093)" index="992" value="0"/>
<variable name="ASSIGN(A001,E094)" index="993" value="0"/>
<variable name="ASSIGN(A001,E095)" index="994" value="0"/>
<variable name="ASSIGN(A001,E096)" index="995" value="0"/>
<variable name="ASSIGN(A001,E097)" index="996" value="0"/>
<variable name="ASSIGN(A001,E098)" index="997" value="0"/>
<variable name="ASSIGN(A001,E099)" index="998" value="0"/>
<variable name="ASSIGN(A001,E100)" index="999" value="0"/>
<variable name="ASSIGN(A002,A001)" index="1000" value="0"/>
<variable name="ASSIGN(A002,A002)" index="1001" value="1"/>
<variable name="ASSIGN(A002,A003)" index="1002" value="0"/>
<variable name="ASSIGN(A002,A004)" index="1003" value="0"/>
<variable name="ASSIGN(A002,A005)" index="1004" value="0"/>
<variable name="ASSIGN(A002,A006)" index="1005" value="0"/>
<variable name="ASSIGN(A002,A007)" index="1006" value="0"/>
<variable name="ASSIGN(A002,A008)" index="1007" value="0"/>
<variable name="ASSIGN(A002,A009)" index="1008" value="0"/>
<variable name="ASSIGN(A002,A010)" index="1009" value="0"/>
<variable name="ASSIGN(A002,A011)" index="1010" value="0"/>
<variable name="ASSIGN(A002,A012)" index="1011" value="0"/>
<variable name="ASSIGN(A002,A013)" index="1012" value="0"/>
<variable name="ASSIGN(A002,A014)" index="1013" value="0"/>
<variable name="ASSIGN(A002,A015)" index="1014" value="0"/>
<variable name="ASSIGN(A002,A016)" index="1015" value="0"/>
<variable name="ASSIGN(A002,A017)" index="1016" value="0"/>
<variable name="ASSIGN(A002,A018)" index="1017" value="0"/>
<variable name="ASSIGN(A002,A019)" index="1018" value="0"/>
<variable name="ASSIGN(A002,A020)" index="1019" value="0"/>
<variable name="ASSIGN(A002,A021)" index="1020" value="0"/>
<variable name="ASSIGN(A002,A022)" index="1021" value="0"/>
<variable name="ASSIGN(A002,A023)" index="1022" value="0"/>
<variable name="ASSIGN(A002,A024)" index="1023" value="0"/>
<variable name="ASSIGN(A002,A025)" index="1024" value="0"/>

<variable name="ASSIGN(A002,A026)" index="1025" value="0"/>
<variable name="ASSIGN(A002,A027)" index="1026" value="0"/>
<variable name="ASSIGN(A002,A028)" index="1027" value="0"/>
<variable name="ASSIGN(A002,A029)" index="1028" value="0"/>
<variable name="ASSIGN(A002,A030)" index="1029" value="0"/>
<variable name="ASSIGN(A002,A031)" index="1030" value="0"/>
<variable name="ASSIGN(A002,A032)" index="1031" value="0"/>
<variable name="ASSIGN(A002,A033)" index="1032" value="0"/>
<variable name="ASSIGN(A002,A034)" index="1033" value="0"/>
<variable name="ASSIGN(A002,A035)" index="1034" value="0"/>
<variable name="ASSIGN(A002,A036)" index="1035" value="0"/>
<variable name="ASSIGN(A002,A037)" index="1036" value="0"/>
<variable name="ASSIGN(A002,A038)" index="1037" value="0"/>
<variable name="ASSIGN(A002,A039)" index="1038" value="0"/>
<variable name="ASSIGN(A002,A040)" index="1039" value="0"/>
<variable name="ASSIGN(A002,A041)" index="1040" value="0"/>
<variable name="ASSIGN(A002,A042)" index="1041" value="0"/>
<variable name="ASSIGN(A002,A043)" index="1042" value="0"/>
<variable name="ASSIGN(A002,A044)" index="1043" value="0"/>
<variable name="ASSIGN(A002,A045)" index="1044" value="0"/>
<variable name="ASSIGN(A002,A046)" index="1045" value="0"/>
<variable name="ASSIGN(A002,A047)" index="1046" value="0"/>
<variable name="ASSIGN(A002,A048)" index="1047" value="0"/>
<variable name="ASSIGN(A002,A049)" index="1048" value="0"/>
<variable name="ASSIGN(A002,A050)" index="1049" value="0"/>
<variable name="ASSIGN(A002,A051)" index="1050" value="0"/>
<variable name="ASSIGN(A002,A052)" index="1051" value="0"/>
<variable name="ASSIGN(A002,A053)" index="1052" value="0"/>
<variable name="ASSIGN(A002,A054)" index="1053" value="0"/>
<variable name="ASSIGN(A002,A055)" index="1054" value="0"/>
<variable name="ASSIGN(A002,A056)" index="1055" value="0"/>
<variable name="ASSIGN(A002,A057)" index="1056" value="0"/>
<variable name="ASSIGN(A002,A058)" index="1057" value="0"/>
<variable name="ASSIGN(A002,A059)" index="1058" value="0"/>
<variable name="ASSIGN(A002,A060)" index="1059" value="0"/>
<variable name="ASSIGN(A002,A061)" index="1060" value="0"/>
<variable name="ASSIGN(A002,A062)" index="1061" value="0"/>
<variable name="ASSIGN(A002,A063)" index="1062" value="0"/>
<variable name="ASSIGN(A002,A064)" index="1063" value="0"/>
<variable name="ASSIGN(A002,A065)" index="1064" value="0"/>
<variable name="ASSIGN(A002,A066)" index="1065" value="0"/>
<variable name="ASSIGN(A002,A067)" index="1066" value="0"/>
<variable name="ASSIGN(A002,A068)" index="1067" value="0"/>
<variable name="ASSIGN(A002,A069)" index="1068" value="0"/>
<variable name="ASSIGN(A002,A070)" index="1069" value="0"/>
<variable name="ASSIGN(A002,A071)" index="1070" value="0"/>
<variable name="ASSIGN(A002,A072)" index="1071" value="0"/>
<variable name="ASSIGN(A002,A073)" index="1072" value="0"/>
<variable name="ASSIGN(A002,A074)" index="1073" value="0"/>
<variable name="ASSIGN(A002,A075)" index="1074" value="0"/>
<variable name="ASSIGN(A002,A076)" index="1075" value="0"/>
<variable name="ASSIGN(A002,A077)" index="1076" value="0"/>
<variable name="ASSIGN(A002,A078)" index="1077" value="0"/>
<variable name="ASSIGN(A002,A079)" index="1078" value="0"/>
<variable name="ASSIGN(A002,A080)" index="1079" value="0"/>
<variable name="ASSIGN(A002,A081)" index="1080" value="0"/>
<variable name="ASSIGN(A002,A082)" index="1081" value="0"/>
<variable name="ASSIGN(A002,A083)" index="1082" value="0"/>
<variable name="ASSIGN(A002,A084)" index="1083" value="0"/>
<variable name="ASSIGN(A002,A085)" index="1084" value="0"/>
<variable name="ASSIGN(A002,A086)" index="1085" value="0"/>
<variable name="ASSIGN(A002,A087)" index="1086" value="0"/>
<variable name="ASSIGN(A002,A088)" index="1087" value="0"/>
<variable name="ASSIGN(A002,A089)" index="1088" value="0"/>
<variable name="ASSIGN(A002,A090)" index="1089" value="0"/>
<variable name="ASSIGN(A002,A091)" index="1090" value="0"/>
<variable name="ASSIGN(A002,A092)" index="1091" value="0"/>
<variable name="ASSIGN(A002,A093)" index="1092" value="0"/>
<variable name="ASSIGN(A002,A094)" index="1093" value="0"/>
<variable name="ASSIGN(A002,A095)" index="1094" value="0"/>
<variable name="ASSIGN(A002,A096)" index="1095" value="0"/>
<variable name="ASSIGN(A002,A097)" index="1096" value="0"/>

<variable name="ASSIGN(A002,A098)" index="1097" value="0"/>
<variable name="ASSIGN(A002,A099)" index="1098" value="0"/>
<variable name="ASSIGN(A002,A100)" index="1099" value="0"/>
<variable name="ASSIGN(A002,B001)" index="1100" value="0"/>
<variable name="ASSIGN(A002,B002)" index="1101" value="0"/>
<variable name="ASSIGN(A002,B003)" index="1102" value="0"/>
<variable name="ASSIGN(A002,B004)" index="1103" value="0"/>
<variable name="ASSIGN(A002,B005)" index="1104" value="0"/>
<variable name="ASSIGN(A002,B006)" index="1105" value="0"/>
<variable name="ASSIGN(A002,B007)" index="1106" value="0"/>
<variable name="ASSIGN(A002,B008)" index="1107" value="0"/>
<variable name="ASSIGN(A002,B009)" index="1108" value="0"/>
<variable name="ASSIGN(A002,B010)" index="1109" value="0"/>
<variable name="ASSIGN(A002,B011)" index="1110" value="0"/>
<variable name="ASSIGN(A002,B012)" index="1111" value="0"/>
<variable name="ASSIGN(A002,B013)" index="1112" value="0"/>
<variable name="ASSIGN(A002,B014)" index="1113" value="0"/>
<variable name="ASSIGN(A002,B015)" index="1114" value="0"/>
<variable name="ASSIGN(A002,B016)" index="1115" value="0"/>
<variable name="ASSIGN(A002,B017)" index="1116" value="0"/>
<variable name="ASSIGN(A002,B018)" index="1117" value="0"/>
<variable name="ASSIGN(A002,B019)" index="1118" value="0"/>
<variable name="ASSIGN(A002,B020)" index="1119" value="0"/>
<variable name="ASSIGN(A002,B021)" index="1120" value="0"/>
<variable name="ASSIGN(A002,B022)" index="1121" value="0"/>
<variable name="ASSIGN(A002,B023)" index="1122" value="0"/>
<variable name="ASSIGN(A002,B024)" index="1123" value="0"/>
<variable name="ASSIGN(A002,B025)" index="1124" value="0"/>
<variable name="ASSIGN(A002,B026)" index="1125" value="0"/>
<variable name="ASSIGN(A002,B027)" index="1126" value="0"/>
<variable name="ASSIGN(A002,B028)" index="1127" value="0"/>
<variable name="ASSIGN(A002,B029)" index="1128" value="0"/>
<variable name="ASSIGN(A002,B030)" index="1129" value="0"/>
<variable name="ASSIGN(A002,B031)" index="1130" value="0"/>
<variable name="ASSIGN(A002,B032)" index="1131" value="0"/>
<variable name="ASSIGN(A002,B033)" index="1132" value="0"/>
<variable name="ASSIGN(A002,B034)" index="1133" value="0"/>
<variable name="ASSIGN(A002,B035)" index="1134" value="0"/>
<variable name="ASSIGN(A002,B036)" index="1135" value="0"/>
<variable name="ASSIGN(A002,B037)" index="1136" value="0"/>
<variable name="ASSIGN(A002,B038)" index="1137" value="0"/>
<variable name="ASSIGN(A002,B039)" index="1138" value="0"/>
<variable name="ASSIGN(A002,B040)" index="1139" value="0"/>
<variable name="ASSIGN(A002,B041)" index="1140" value="0"/>
<variable name="ASSIGN(A002,B042)" index="1141" value="0"/>
<variable name="ASSIGN(A002,B043)" index="1142" value="0"/>
<variable name="ASSIGN(A002,B044)" index="1143" value="0"/>
<variable name="ASSIGN(A002,B045)" index="1144" value="0"/>
<variable name="ASSIGN(A002,B046)" index="1145" value="0"/>
<variable name="ASSIGN(A002,B047)" index="1146" value="0"/>
<variable name="ASSIGN(A002,B048)" index="1147" value="0"/>
<variable name="ASSIGN(A002,B049)" index="1148" value="0"/>
<variable name="ASSIGN(A002,B050)" index="1149" value="0"/>
<variable name="ASSIGN(A002,B051)" index="1150" value="0"/>
<variable name="ASSIGN(A002,B052)" index="1151" value="0"/>
<variable name="ASSIGN(A002,B053)" index="1152" value="0"/>
<variable name="ASSIGN(A002,B054)" index="1153" value="0"/>
<variable name="ASSIGN(A002,B055)" index="1154" value="0"/>
<variable name="ASSIGN(A002,B056)" index="1155" value="0"/>
<variable name="ASSIGN(A002,B057)" index="1156" value="0"/>
<variable name="ASSIGN(A002,B058)" index="1157" value="0"/>
<variable name="ASSIGN(A002,B059)" index="1158" value="0"/>
<variable name="ASSIGN(A002,B060)" index="1159" value="0"/>
<variable name="ASSIGN(A002,B061)" index="1160" value="0"/>
<variable name="ASSIGN(A002,B062)" index="1161" value="0"/>
<variable name="ASSIGN(A002,B063)" index="1162" value="0"/>
<variable name="ASSIGN(A002,B064)" index="1163" value="0"/>
<variable name="ASSIGN(A002,B065)" index="1164" value="0"/>
<variable name="ASSIGN(A002,B066)" index="1165" value="0"/>
<variable name="ASSIGN(A002,B067)" index="1166" value="0"/>
<variable name="ASSIGN(A002,B068)" index="1167" value="0"/>
<variable name="ASSIGN(A002,B069)" index="1168" value="0"/>

<variable name="ASSIGN(A002,B070)" index="1169" value="0"/>
<variable name="ASSIGN(A002,B071)" index="1170" value="0"/>
<variable name="ASSIGN(A002,B072)" index="1171" value="0"/>
<variable name="ASSIGN(A002,B073)" index="1172" value="0"/>
<variable name="ASSIGN(A002,B074)" index="1173" value="0"/>
<variable name="ASSIGN(A002,B075)" index="1174" value="0"/>
<variable name="ASSIGN(A002,B076)" index="1175" value="0"/>
<variable name="ASSIGN(A002,B077)" index="1176" value="0"/>
<variable name="ASSIGN(A002,B078)" index="1177" value="0"/>
<variable name="ASSIGN(A002,B079)" index="1178" value="0"/>
<variable name="ASSIGN(A002,B080)" index="1179" value="0"/>
<variable name="ASSIGN(A002,B081)" index="1180" value="0"/>
<variable name="ASSIGN(A002,B082)" index="1181" value="0"/>
<variable name="ASSIGN(A002,B083)" index="1182" value="0"/>
<variable name="ASSIGN(A002,B084)" index="1183" value="0"/>
<variable name="ASSIGN(A002,B085)" index="1184" value="0"/>
<variable name="ASSIGN(A002,B086)" index="1185" value="0"/>
<variable name="ASSIGN(A002,B087)" index="1186" value="0"/>
<variable name="ASSIGN(A002,B088)" index="1187" value="0"/>
<variable name="ASSIGN(A002,B089)" index="1188" value="0"/>
<variable name="ASSIGN(A002,B090)" index="1189" value="0"/>
<variable name="ASSIGN(A002,B091)" index="1190" value="0"/>
<variable name="ASSIGN(A002,B092)" index="1191" value="0"/>
<variable name="ASSIGN(A002,B093)" index="1192" value="0"/>
<variable name="ASSIGN(A002,B094)" index="1193" value="0"/>
<variable name="ASSIGN(A002,B095)" index="1194" value="0"/>
<variable name="ASSIGN(A002,B096)" index="1195" value="0"/>
<variable name="ASSIGN(A002,B097)" index="1196" value="0"/>
<variable name="ASSIGN(A002,B098)" index="1197" value="0"/>
<variable name="ASSIGN(A002,B099)" index="1198" value="0"/>
<variable name="ASSIGN(A002,B100)" index="1199" value="0"/>
<variable name="ASSIGN(A002,C001)" index="1200" value="0"/>
<variable name="ASSIGN(A002,C002)" index="1201" value="0"/>
<variable name="ASSIGN(A002,C003)" index="1202" value="0"/>
<variable name="ASSIGN(A002,C004)" index="1203" value="0"/>
<variable name="ASSIGN(A002,C005)" index="1204" value="0"/>
<variable name="ASSIGN(A002,C006)" index="1205" value="0"/>
<variable name="ASSIGN(A002,C007)" index="1206" value="0"/>
<variable name="ASSIGN(A002,C008)" index="1207" value="0"/>
<variable name="ASSIGN(A002,C009)" index="1208" value="0"/>
<variable name="ASSIGN(A002,C010)" index="1209" value="0"/>
<variable name="ASSIGN(A002,C011)" index="1210" value="0"/>
<variable name="ASSIGN(A002,C012)" index="1211" value="0"/>
<variable name="ASSIGN(A002,C013)" index="1212" value="0"/>
<variable name="ASSIGN(A002,C014)" index="1213" value="0"/>
<variable name="ASSIGN(A002,C015)" index="1214" value="0"/>
<variable name="ASSIGN(A002,C016)" index="1215" value="0"/>
<variable name="ASSIGN(A002,C017)" index="1216" value="0"/>
<variable name="ASSIGN(A002,C018)" index="1217" value="0"/>
<variable name="ASSIGN(A002,C019)" index="1218" value="0"/>
<variable name="ASSIGN(A002,C020)" index="1219" value="0"/>
<variable name="ASSIGN(A002,C021)" index="1220" value="0"/>
<variable name="ASSIGN(A002,C022)" index="1221" value="0"/>
<variable name="ASSIGN(A002,C023)" index="1222" value="0"/>
<variable name="ASSIGN(A002,C024)" index="1223" value="0"/>
<variable name="ASSIGN(A002,C025)" index="1224" value="0"/>
<variable name="ASSIGN(A002,C026)" index="1225" value="0"/>
<variable name="ASSIGN(A002,C027)" index="1226" value="0"/>
<variable name="ASSIGN(A002,C028)" index="1227" value="0"/>
<variable name="ASSIGN(A002,C029)" index="1228" value="0"/>
<variable name="ASSIGN(A002,C030)" index="1229" value="0"/>
<variable name="ASSIGN(A002,C031)" index="1230" value="0"/>
<variable name="ASSIGN(A002,C032)" index="1231" value="0"/>
<variable name="ASSIGN(A002,C033)" index="1232" value="0"/>
<variable name="ASSIGN(A002,C034)" index="1233" value="0"/>
<variable name="ASSIGN(A002,C035)" index="1234" value="0"/>
<variable name="ASSIGN(A002,C036)" index="1235" value="0"/>
<variable name="ASSIGN(A002,C037)" index="1236" value="0"/>
<variable name="ASSIGN(A002,C038)" index="1237" value="0"/>
<variable name="ASSIGN(A002,C039)" index="1238" value="0"/>
<variable name="ASSIGN(A002,C040)" index="1239" value="0"/>
<variable name="ASSIGN(A002,C041)" index="1240" value="0"/>

<variable name="ASSIGN(A002,C042)" index="1241" value="0"/>
<variable name="ASSIGN(A002,C043)" index="1242" value="0"/>
<variable name="ASSIGN(A002,C044)" index="1243" value="0"/>
<variable name="ASSIGN(A002,C045)" index="1244" value="0"/>
<variable name="ASSIGN(A002,C046)" index="1245" value="0"/>
<variable name="ASSIGN(A002,C047)" index="1246" value="0"/>
<variable name="ASSIGN(A002,C048)" index="1247" value="0"/>
<variable name="ASSIGN(A002,C049)" index="1248" value="0"/>
<variable name="ASSIGN(A002,C050)" index="1249" value="0"/>
<variable name="ASSIGN(A002,C051)" index="1250" value="0"/>
<variable name="ASSIGN(A002,C052)" index="1251" value="0"/>
<variable name="ASSIGN(A002,C053)" index="1252" value="0"/>
<variable name="ASSIGN(A002,C054)" index="1253" value="0"/>
<variable name="ASSIGN(A002,C055)" index="1254" value="0"/>
<variable name="ASSIGN(A002,C056)" index="1255" value="0"/>
<variable name="ASSIGN(A002,C057)" index="1256" value="0"/>
<variable name="ASSIGN(A002,C058)" index="1257" value="0"/>
<variable name="ASSIGN(A002,C059)" index="1258" value="0"/>
<variable name="ASSIGN(A002,C060)" index="1259" value="0"/>
<variable name="ASSIGN(A002,C061)" index="1260" value="0"/>
<variable name="ASSIGN(A002,C062)" index="1261" value="0"/>
<variable name="ASSIGN(A002,C063)" index="1262" value="0"/>
<variable name="ASSIGN(A002,C064)" index="1263" value="0"/>
<variable name="ASSIGN(A002,C065)" index="1264" value="0"/>
<variable name="ASSIGN(A002,C066)" index="1265" value="0"/>
<variable name="ASSIGN(A002,C067)" index="1266" value="0"/>
<variable name="ASSIGN(A002,C068)" index="1267" value="0"/>
<variable name="ASSIGN(A002,C069)" index="1268" value="0"/>
<variable name="ASSIGN(A002,C070)" index="1269" value="0"/>
<variable name="ASSIGN(A002,C071)" index="1270" value="0"/>
<variable name="ASSIGN(A002,C072)" index="1271" value="0"/>
<variable name="ASSIGN(A002,C073)" index="1272" value="0"/>
<variable name="ASSIGN(A002,C074)" index="1273" value="0"/>
<variable name="ASSIGN(A002,C075)" index="1274" value="0"/>
<variable name="ASSIGN(A002,C076)" index="1275" value="0"/>
<variable name="ASSIGN(A002,C077)" index="1276" value="0"/>
<variable name="ASSIGN(A002,C078)" index="1277" value="0"/>
<variable name="ASSIGN(A002,C079)" index="1278" value="0"/>
<variable name="ASSIGN(A002,C080)" index="1279" value="0"/>
<variable name="ASSIGN(A002,C081)" index="1280" value="0"/>
<variable name="ASSIGN(A002,C082)" index="1281" value="0"/>
<variable name="ASSIGN(A002,C083)" index="1282" value="0"/>
<variable name="ASSIGN(A002,C084)" index="1283" value="0"/>
<variable name="ASSIGN(A002,C085)" index="1284" value="0"/>
<variable name="ASSIGN(A002,C086)" index="1285" value="0"/>
<variable name="ASSIGN(A002,C087)" index="1286" value="0"/>
<variable name="ASSIGN(A002,C088)" index="1287" value="0"/>
<variable name="ASSIGN(A002,C089)" index="1288" value="0"/>
<variable name="ASSIGN(A002,C090)" index="1289" value="0"/>
<variable name="ASSIGN(A002,C091)" index="1290" value="0"/>
<variable name="ASSIGN(A002,C092)" index="1291" value="0"/>
<variable name="ASSIGN(A002,C093)" index="1292" value="0"/>
<variable name="ASSIGN(A002,C094)" index="1293" value="0"/>
<variable name="ASSIGN(A002,C095)" index="1294" value="0"/>
<variable name="ASSIGN(A002,C096)" index="1295" value="0"/>
<variable name="ASSIGN(A002,C097)" index="1296" value="0"/>
<variable name="ASSIGN(A002,C098)" index="1297" value="0"/>
<variable name="ASSIGN(A002,C099)" index="1298" value="0"/>
<variable name="ASSIGN(A002,C100)" index="1299" value="0"/>
<variable name="ASSIGN(A002,D001)" index="1300" value="0"/>
<variable name="ASSIGN(A002,D002)" index="1301" value="0"/>
<variable name="ASSIGN(A002,D003)" index="1302" value="0"/>
<variable name="ASSIGN(A002,D004)" index="1303" value="0"/>
<variable name="ASSIGN(A002,D005)" index="1304" value="0"/>
<variable name="ASSIGN(A002,D006)" index="1305" value="0"/>
<variable name="ASSIGN(A002,D007)" index="1306" value="0"/>
<variable name="ASSIGN(A002,D008)" index="1307" value="0"/>
<variable name="ASSIGN(A002,D009)" index="1308" value="0"/>
<variable name="ASSIGN(A002,D010)" index="1309" value="0"/>
<variable name="ASSIGN(A002,D011)" index="1310" value="0"/>
<variable name="ASSIGN(A002,D012)" index="1311" value="0"/>
<variable name="ASSIGN(A002,D013)" index="1312" value="0"/>

<variable name="ASSIGN(A002,D014)" index="1313" value="0"/>
<variable name="ASSIGN(A002,D015)" index="1314" value="0"/>
<variable name="ASSIGN(A002,D016)" index="1315" value="0"/>
<variable name="ASSIGN(A002,D017)" index="1316" value="0"/>
<variable name="ASSIGN(A002,D018)" index="1317" value="0"/>
<variable name="ASSIGN(A002,D019)" index="1318" value="0"/>
<variable name="ASSIGN(A002,D020)" index="1319" value="0"/>
<variable name="ASSIGN(A002,D021)" index="1320" value="0"/>
<variable name="ASSIGN(A002,D022)" index="1321" value="0"/>
<variable name="ASSIGN(A002,D023)" index="1322" value="0"/>
<variable name="ASSIGN(A002,D024)" index="1323" value="0"/>
<variable name="ASSIGN(A002,D025)" index="1324" value="0"/>
<variable name="ASSIGN(A002,D026)" index="1325" value="0"/>
<variable name="ASSIGN(A002,D027)" index="1326" value="0"/>
<variable name="ASSIGN(A002,D028)" index="1327" value="0"/>
<variable name="ASSIGN(A002,D029)" index="1328" value="0"/>
<variable name="ASSIGN(A002,D030)" index="1329" value="0"/>
<variable name="ASSIGN(A002,D031)" index="1330" value="0"/>
<variable name="ASSIGN(A002,D032)" index="1331" value="0"/>
<variable name="ASSIGN(A002,D033)" index="1332" value="0"/>
<variable name="ASSIGN(A002,D034)" index="1333" value="0"/>
<variable name="ASSIGN(A002,D035)" index="1334" value="0"/>
<variable name="ASSIGN(A002,D036)" index="1335" value="0"/>
<variable name="ASSIGN(A002,D037)" index="1336" value="0"/>
<variable name="ASSIGN(A002,D038)" index="1337" value="0"/>
<variable name="ASSIGN(A002,D039)" index="1338" value="0"/>
<variable name="ASSIGN(A002,D040)" index="1339" value="0"/>
<variable name="ASSIGN(A002,D041)" index="1340" value="0"/>
<variable name="ASSIGN(A002,D042)" index="1341" value="0"/>
<variable name="ASSIGN(A002,D043)" index="1342" value="0"/>
<variable name="ASSIGN(A002,D044)" index="1343" value="0"/>
<variable name="ASSIGN(A002,D045)" index="1344" value="0"/>
<variable name="ASSIGN(A002,D046)" index="1345" value="0"/>
<variable name="ASSIGN(A002,D047)" index="1346" value="0"/>
<variable name="ASSIGN(A002,D048)" index="1347" value="0"/>
<variable name="ASSIGN(A002,D049)" index="1348" value="0"/>
<variable name="ASSIGN(A002,D050)" index="1349" value="0"/>
<variable name="ASSIGN(A002,D051)" index="1350" value="0"/>
<variable name="ASSIGN(A002,D052)" index="1351" value="0"/>
<variable name="ASSIGN(A002,D053)" index="1352" value="0"/>
<variable name="ASSIGN(A002,D054)" index="1353" value="0"/>
<variable name="ASSIGN(A002,D055)" index="1354" value="0"/>
<variable name="ASSIGN(A002,D056)" index="1355" value="0"/>
<variable name="ASSIGN(A002,D057)" index="1356" value="0"/>
<variable name="ASSIGN(A002,D058)" index="1357" value="0"/>
<variable name="ASSIGN(A002,D059)" index="1358" value="0"/>
<variable name="ASSIGN(A002,D060)" index="1359" value="0"/>
<variable name="ASSIGN(A002,D061)" index="1360" value="0"/>
<variable name="ASSIGN(A002,D062)" index="1361" value="0"/>
<variable name="ASSIGN(A002,D063)" index="1362" value="0"/>
<variable name="ASSIGN(A002,D064)" index="1363" value="0"/>
<variable name="ASSIGN(A002,D065)" index="1364" value="0"/>
<variable name="ASSIGN(A002,D066)" index="1365" value="0"/>
<variable name="ASSIGN(A002,D067)" index="1366" value="0"/>
<variable name="ASSIGN(A002,D068)" index="1367" value="0"/>
<variable name="ASSIGN(A002,D069)" index="1368" value="0"/>
<variable name="ASSIGN(A002,D070)" index="1369" value="0"/>
<variable name="ASSIGN(A002,D071)" index="1370" value="0"/>
<variable name="ASSIGN(A002,D072)" index="1371" value="0"/>
<variable name="ASSIGN(A002,D073)" index="1372" value="0"/>
<variable name="ASSIGN(A002,D074)" index="1373" value="0"/>
<variable name="ASSIGN(A002,D075)" index="1374" value="0"/>
<variable name="ASSIGN(A002,D076)" index="1375" value="0"/>
<variable name="ASSIGN(A002,D077)" index="1376" value="0"/>
<variable name="ASSIGN(A002,D078)" index="1377" value="0"/>
<variable name="ASSIGN(A002,D079)" index="1378" value="0"/>
<variable name="ASSIGN(A002,D080)" index="1379" value="0"/>
<variable name="ASSIGN(A002,D081)" index="1380" value="0"/>
<variable name="ASSIGN(A002,D082)" index="1381" value="0"/>
<variable name="ASSIGN(A002,D083)" index="1382" value="0"/>
<variable name="ASSIGN(A002,D084)" index="1383" value="0"/>
<variable name="ASSIGN(A002,D085)" index="1384" value="0"/>

<variable name="ASSIGN(A002,D086)" index="1385" value="0"/>
<variable name="ASSIGN(A002,D087)" index="1386" value="0"/>
<variable name="ASSIGN(A002,D088)" index="1387" value="0"/>
<variable name="ASSIGN(A002,D089)" index="1388" value="0"/>
<variable name="ASSIGN(A002,D090)" index="1389" value="0"/>
<variable name="ASSIGN(A002,D091)" index="1390" value="0"/>
<variable name="ASSIGN(A002,D092)" index="1391" value="0"/>
<variable name="ASSIGN(A002,D093)" index="1392" value="0"/>
<variable name="ASSIGN(A002,D094)" index="1393" value="0"/>
<variable name="ASSIGN(A002,D095)" index="1394" value="0"/>
<variable name="ASSIGN(A002,D096)" index="1395" value="0"/>
<variable name="ASSIGN(A002,D097)" index="1396" value="0"/>
<variable name="ASSIGN(A002,D098)" index="1397" value="0"/>
<variable name="ASSIGN(A002,D099)" index="1398" value="0"/>
<variable name="ASSIGN(A002,D100)" index="1399" value="0"/>
<variable name="ASSIGN(A002,E001)" index="1400" value="0"/>
<variable name="ASSIGN(A002,E002)" index="1401" value="0"/>
<variable name="ASSIGN(A002,E003)" index="1402" value="0"/>
<variable name="ASSIGN(A002,E004)" index="1403" value="0"/>
<variable name="ASSIGN(A002,E005)" index="1404" value="0"/>
<variable name="ASSIGN(A002,E006)" index="1405" value="0"/>
<variable name="ASSIGN(A002,E007)" index="1406" value="0"/>
<variable name="ASSIGN(A002,E008)" index="1407" value="0"/>
<variable name="ASSIGN(A002,E009)" index="1408" value="0"/>
<variable name="ASSIGN(A002,E010)" index="1409" value="0"/>
<variable name="ASSIGN(A002,E011)" index="1410" value="0"/>
<variable name="ASSIGN(A002,E012)" index="1411" value="0"/>
<variable name="ASSIGN(A002,E013)" index="1412" value="0"/>
<variable name="ASSIGN(A002,E014)" index="1413" value="0"/>
<variable name="ASSIGN(A002,E015)" index="1414" value="0"/>
<variable name="ASSIGN(A002,E016)" index="1415" value="0"/>
<variable name="ASSIGN(A002,E017)" index="1416" value="0"/>
<variable name="ASSIGN(A002,E018)" index="1417" value="0"/>
<variable name="ASSIGN(A002,E019)" index="1418" value="0"/>
<variable name="ASSIGN(A002,E020)" index="1419" value="0"/>
<variable name="ASSIGN(A002,E021)" index="1420" value="0"/>
<variable name="ASSIGN(A002,E022)" index="1421" value="0"/>
<variable name="ASSIGN(A002,E023)" index="1422" value="0"/>
<variable name="ASSIGN(A002,E024)" index="1423" value="0"/>
<variable name="ASSIGN(A002,E025)" index="1424" value="0"/>
<variable name="ASSIGN(A002,E026)" index="1425" value="0"/>
<variable name="ASSIGN(A002,E027)" index="1426" value="0"/>
<variable name="ASSIGN(A002,E028)" index="1427" value="0"/>
<variable name="ASSIGN(A002,E029)" index="1428" value="0"/>
<variable name="ASSIGN(A002,E030)" index="1429" value="0"/>
<variable name="ASSIGN(A002,E031)" index="1430" value="0"/>
<variable name="ASSIGN(A002,E032)" index="1431" value="0"/>
<variable name="ASSIGN(A002,E033)" index="1432" value="0"/>
<variable name="ASSIGN(A002,E034)" index="1433" value="0"/>
<variable name="ASSIGN(A002,E035)" index="1434" value="0"/>
<variable name="ASSIGN(A002,E036)" index="1435" value="0"/>
<variable name="ASSIGN(A002,E037)" index="1436" value="0"/>
<variable name="ASSIGN(A002,E038)" index="1437" value="0"/>
<variable name="ASSIGN(A002,E039)" index="1438" value="0"/>
<variable name="ASSIGN(A002,E040)" index="1439" value="0"/>
<variable name="ASSIGN(A002,E041)" index="1440" value="0"/>
<variable name="ASSIGN(A002,E042)" index="1441" value="0"/>
<variable name="ASSIGN(A002,E043)" index="1442" value="0"/>
<variable name="ASSIGN(A002,E044)" index="1443" value="0"/>
<variable name="ASSIGN(A002,E045)" index="1444" value="0"/>
<variable name="ASSIGN(A002,E046)" index="1445" value="0"/>
<variable name="ASSIGN(A002,E047)" index="1446" value="0"/>
<variable name="ASSIGN(A002,E048)" index="1447" value="0"/>
<variable name="ASSIGN(A002,E049)" index="1448" value="0"/>
<variable name="ASSIGN(A002,E050)" index="1449" value="0"/>
<variable name="ASSIGN(A002,E051)" index="1450" value="0"/>
<variable name="ASSIGN(A002,E052)" index="1451" value="0"/>
<variable name="ASSIGN(A002,E053)" index="1452" value="0"/>
<variable name="ASSIGN(A002,E054)" index="1453" value="0"/>
<variable name="ASSIGN(A002,E055)" index="1454" value="0"/>
<variable name="ASSIGN(A002,E056)" index="1455" value="0"/>
<variable name="ASSIGN(A002,E057)" index="1456" value="0"/>

<variable name="ASSIGN(A002,E058)" index="1457" value="0"/>
<variable name="ASSIGN(A002,E059)" index="1458" value="0"/>
<variable name="ASSIGN(A002,E060)" index="1459" value="0"/>
<variable name="ASSIGN(A002,E061)" index="1460" value="0"/>
<variable name="ASSIGN(A002,E062)" index="1461" value="0"/>
<variable name="ASSIGN(A002,E063)" index="1462" value="0"/>
<variable name="ASSIGN(A002,E064)" index="1463" value="0"/>
<variable name="ASSIGN(A002,E065)" index="1464" value="0"/>
<variable name="ASSIGN(A002,E066)" index="1465" value="0"/>
<variable name="ASSIGN(A002,E067)" index="1466" value="0"/>
<variable name="ASSIGN(A002,E068)" index="1467" value="0"/>
<variable name="ASSIGN(A002,E069)" index="1468" value="0"/>
<variable name="ASSIGN(A002,E070)" index="1469" value="0"/>
<variable name="ASSIGN(A002,E071)" index="1470" value="0"/>
<variable name="ASSIGN(A002,E072)" index="1471" value="0"/>
<variable name="ASSIGN(A002,E073)" index="1472" value="0"/>
<variable name="ASSIGN(A002,E074)" index="1473" value="0"/>
<variable name="ASSIGN(A002,E075)" index="1474" value="0"/>
<variable name="ASSIGN(A002,E076)" index="1475" value="0"/>
<variable name="ASSIGN(A002,E077)" index="1476" value="0"/>
<variable name="ASSIGN(A002,E078)" index="1477" value="0"/>
<variable name="ASSIGN(A002,E079)" index="1478" value="0"/>
<variable name="ASSIGN(A002,E080)" index="1479" value="0"/>
<variable name="ASSIGN(A002,E081)" index="1480" value="0"/>
<variable name="ASSIGN(A002,E082)" index="1481" value="0"/>
<variable name="ASSIGN(A002,E083)" index="1482" value="0"/>
<variable name="ASSIGN(A002,E084)" index="1483" value="0"/>
<variable name="ASSIGN(A002,E085)" index="1484" value="0"/>
<variable name="ASSIGN(A002,E086)" index="1485" value="0"/>
<variable name="ASSIGN(A002,E087)" index="1486" value="0"/>
<variable name="ASSIGN(A002,E088)" index="1487" value="0"/>
<variable name="ASSIGN(A002,E089)" index="1488" value="0"/>
<variable name="ASSIGN(A002,E090)" index="1489" value="0"/>
<variable name="ASSIGN(A002,E091)" index="1490" value="0"/>
<variable name="ASSIGN(A002,E092)" index="1491" value="0"/>
<variable name="ASSIGN(A002,E093)" index="1492" value="0"/>
<variable name="ASSIGN(A002,E094)" index="1493" value="0"/>
<variable name="ASSIGN(A002,E095)" index="1494" value="0"/>
<variable name="ASSIGN(A002,E096)" index="1495" value="0"/>
<variable name="ASSIGN(A002,E097)" index="1496" value="0"/>
<variable name="ASSIGN(A002,E098)" index="1497" value="0"/>
<variable name="ASSIGN(A002,E099)" index="1498" value="0"/>
<variable name="ASSIGN(A002,E100)" index="1499" value="0"/>

3.3 Output for Allocation-handling Variables

<variable name="ALLOCATE(X,A001)" index="250500" value="0"/>
<variable name="ALLOCATE(X,A002)" index="250501" value="0"/>
<variable name="ALLOCATE(X,A003)" index="250502" value="0"/>
<variable name="ALLOCATE(X,A004)" index="250503" value="0"/>
<variable name="ALLOCATE(X,A005)" index="250504" value="0"/>
<variable name="ALLOCATE(X,A006)" index="250505" value="0"/>
<variable name="ALLOCATE(X,A007)" index="250506" value="0"/>
<variable name="ALLOCATE(X,A008)" index="250507" value="1"/>
<variable name="ALLOCATE(X,A009)" index="250508" value="0"/>
<variable name="ALLOCATE(X,A010)" index="250509" value="0"/>
<variable name="ALLOCATE(X,A011)" index="250510" value="0"/>
<variable name="ALLOCATE(X,A012)" index="250511" value="0"/>
<variable name="ALLOCATE(X,A013)" index="250512" value="0"/>
<variable name="ALLOCATE(X,A014)" index="250513" value="0"/>
<variable name="ALLOCATE(X,A015)" index="250514" value="0"/>
<variable name="ALLOCATE(X,A016)" index="250515" value="0"/>
<variable name="ALLOCATE(X,A017)" index="250516" value="0"/>
<variable name="ALLOCATE(X,A018)" index="250517" value="0"/>
<variable name="ALLOCATE(X,A019)" index="250518" value="0"/>
<variable name="ALLOCATE(X,A020)" index="250519" value="0"/>
<variable name="ALLOCATE(X,A021)" index="250520" value="0"/>
<variable name="ALLOCATE(X,A022)" index="250521" value="0"/>

<variable name="ALLOCATE(X,A023)" index="250522" value="0"/>
<variable name="ALLOCATE(X,A024)" index="250523" value="0"/>
<variable name="ALLOCATE(X,A025)" index="250524" value="0"/>
<variable name="ALLOCATE(X,A026)" index="250525" value="0"/>
<variable name="ALLOCATE(X,A027)" index="250526" value="0"/>
<variable name="ALLOCATE(X,A028)" index="250527" value="0"/>
<variable name="ALLOCATE(X,A029)" index="250528" value="0"/>
<variable name="ALLOCATE(X,A030)" index="250529" value="0"/>
<variable name="ALLOCATE(X,A031)" index="250530" value="0"/>
<variable name="ALLOCATE(X,A032)" index="250531" value="0"/>
<variable name="ALLOCATE(X,A033)" index="250532" value="0"/>
<variable name="ALLOCATE(X,A034)" index="250533" value="0"/>
<variable name="ALLOCATE(X,A035)" index="250534" value="0"/>
<variable name="ALLOCATE(X,A036)" index="250535" value="0"/>
<variable name="ALLOCATE(X,A037)" index="250536" value="0"/>
<variable name="ALLOCATE(X,A038)" index="250537" value="0"/>
<variable name="ALLOCATE(X,A039)" index="250538" value="1"/>
<variable name="ALLOCATE(X,A040)" index="250539" value="0"/>
<variable name="ALLOCATE(X,A041)" index="250540" value="1"/>
<variable name="ALLOCATE(X,A042)" index="250541" value="0"/>
<variable name="ALLOCATE(X,A043)" index="250542" value="0"/>
<variable name="ALLOCATE(X,A044)" index="250543" value="0"/>
<variable name="ALLOCATE(X,A045)" index="250544" value="0"/>
<variable name="ALLOCATE(X,A046)" index="250545" value="0"/>
<variable name="ALLOCATE(X,A047)" index="250546" value="0"/>
<variable name="ALLOCATE(X,A048)" index="250547" value="0"/>
<variable name="ALLOCATE(X,A049)" index="250548" value="0"/>
<variable name="ALLOCATE(X,A050)" index="250549" value="1"/>
<variable name="ALLOCATE(X,A051)" index="250550" value="0"/>
<variable name="ALLOCATE(X,A052)" index="250551" value="0"/>
<variable name="ALLOCATE(X,A053)" index="250552" value="0"/>
<variable name="ALLOCATE(X,A054)" index="250553" value="1"/>
<variable name="ALLOCATE(X,A055)" index="250554" value="0"/>
<variable name="ALLOCATE(X,A056)" index="250555" value="0"/>
<variable name="ALLOCATE(X,A057)" index="250556" value="0"/>
<variable name="ALLOCATE(X,A058)" index="250557" value="0"/>
<variable name="ALLOCATE(X,A059)" index="250558" value="0"/>
<variable name="ALLOCATE(X,A060)" index="250559" value="0"/>
<variable name="ALLOCATE(X,A061)" index="250560" value="0"/>
<variable name="ALLOCATE(X,A062)" index="250561" value="1"/>
<variable name="ALLOCATE(X,A063)" index="250562" value="0"/>
<variable name="ALLOCATE(X,A064)" index="250563" value="0"/>
<variable name="ALLOCATE(X,A065)" index="250564" value="0"/>
<variable name="ALLOCATE(X,A066)" index="250565" value="0"/>
<variable name="ALLOCATE(X,A067)" index="250566" value="1"/>
<variable name="ALLOCATE(X,A068)" index="250567" value="0"/>
<variable name="ALLOCATE(X,A069)" index="250568" value="1"/>
<variable name="ALLOCATE(X,A070)" index="250569" value="1"/>
<variable name="ALLOCATE(X,A071)" index="250570" value="0"/>
<variable name="ALLOCATE(X,A072)" index="250571" value="0"/>
<variable name="ALLOCATE(X,A073)" index="250572" value="0"/>
<variable name="ALLOCATE(X,A074)" index="250573" value="0"/>
<variable name="ALLOCATE(X,A075)" index="250574" value="0"/>
<variable name="ALLOCATE(X,A076)" index="250575" value="0"/>
<variable name="ALLOCATE(X,A077)" index="250576" value="0"/>
<variable name="ALLOCATE(X,A078)" index="250577" value="0"/>
<variable name="ALLOCATE(X,A079)" index="250578" value="0"/>
<variable name="ALLOCATE(X,A080)" index="250579" value="0"/>
<variable name="ALLOCATE(X,A081)" index="250580" value="1"/>
<variable name="ALLOCATE(X,A082)" index="250581" value="0"/>
<variable name="ALLOCATE(X,A083)" index="250582" value="0"/>
<variable name="ALLOCATE(X,A084)" index="250583" value="0"/>
<variable name="ALLOCATE(X,A085)" index="250584" value="0"/>
<variable name="ALLOCATE(X,A086)" index="250585" value="0"/>
<variable name="ALLOCATE(X,A087)" index="250586" value="0"/>
<variable name="ALLOCATE(X,A088)" index="250587" value="1"/>
<variable name="ALLOCATE(X,A089)" index="250588" value="0"/>
<variable name="ALLOCATE(X,A090)" index="250589" value="0"/>
<variable name="ALLOCATE(X,A091)" index="250590" value="0"/>
<variable name="ALLOCATE(X,A092)" index="250591" value="0"/>
<variable name="ALLOCATE(X,A093)" index="250592" value="0"/>
<variable name="ALLOCATE(X,A094)" index="250593" value="0"/>

<variable name="ALLOCATE(X,A095)" index="250594" value="0"/>
<variable name="ALLOCATE(X,A096)" index="250595" value="0"/>
<variable name="ALLOCATE(X,A097)" index="250596" value="0"/>
<variable name="ALLOCATE(X,A098)" index="250597" value="1"/>
<variable name="ALLOCATE(X,A099)" index="250598" value="0"/>
<variable name="ALLOCATE(X,A100)" index="250599" value="0"/>
<variable name="ALLOCATE(X,B001)" index="250600" value="0"/>
<variable name="ALLOCATE(X,B002)" index="250601" value="0"/>
<variable name="ALLOCATE(X,B003)" index="250602" value="0"/>
<variable name="ALLOCATE(X,B004)" index="250603" value="1"/>
<variable name="ALLOCATE(X,B005)" index="250604" value="0"/>
<variable name="ALLOCATE(X,B006)" index="250605" value="0"/>
<variable name="ALLOCATE(X,B007)" index="250606" value="0"/>
<variable name="ALLOCATE(X,B008)" index="250607" value="0"/>
<variable name="ALLOCATE(X,B009)" index="250608" value="0"/>
<variable name="ALLOCATE(X,B010)" index="250609" value="1"/>
<variable name="ALLOCATE(X,B011)" index="250610" value="0"/>
<variable name="ALLOCATE(X,B012)" index="250611" value="0"/>
<variable name="ALLOCATE(X,B013)" index="250612" value="0"/>
<variable name="ALLOCATE(X,B014)" index="250613" value="0"/>
<variable name="ALLOCATE(X,B015)" index="250614" value="0"/>
<variable name="ALLOCATE(X,B016)" index="250615" value="0"/>
<variable name="ALLOCATE(X,B017)" index="250616" value="0"/>
<variable name="ALLOCATE(X,B018)" index="250617" value="0"/>
<variable name="ALLOCATE(X,B019)" index="250618" value="0"/>
<variable name="ALLOCATE(X,B020)" index="250619" value="0"/>
<variable name="ALLOCATE(X,B021)" index="250620" value="0"/>
<variable name="ALLOCATE(X,B022)" index="250621" value="0"/>
<variable name="ALLOCATE(X,B023)" index="250622" value="0"/>
<variable name="ALLOCATE(X,B024)" index="250623" value="0"/>
<variable name="ALLOCATE(X,B025)" index="250624" value="0"/>
<variable name="ALLOCATE(X,B026)" index="250625" value="0"/>
<variable name="ALLOCATE(X,B027)" index="250626" value="0"/>
<variable name="ALLOCATE(X,B028)" index="250627" value="0"/>
<variable name="ALLOCATE(X,B029)" index="250628" value="1"/>
<variable name="ALLOCATE(X,B030)" index="250629" value="0"/>
<variable name="ALLOCATE(X,B031)" index="250630" value="0"/>
<variable name="ALLOCATE(X,B032)" index="250631" value="0"/>
<variable name="ALLOCATE(X,B033)" index="250632" value="0"/>
<variable name="ALLOCATE(X,B034)" index="250633" value="0"/>
<variable name="ALLOCATE(X,B035)" index="250634" value="0"/>
<variable name="ALLOCATE(X,B036)" index="250635" value="0"/>
<variable name="ALLOCATE(X,B037)" index="250636" value="0"/>
<variable name="ALLOCATE(X,B038)" index="250637" value="0"/>
<variable name="ALLOCATE(X,B039)" index="250638" value="0"/>
<variable name="ALLOCATE(X,B040)" index="250639" value="0"/>
<variable name="ALLOCATE(X,B041)" index="250640" value="0"/>
<variable name="ALLOCATE(X,B042)" index="250641" value="0"/>
<variable name="ALLOCATE(X,B043)" index="250642" value="0"/>
<variable name="ALLOCATE(X,B044)" index="250643" value="0"/>
<variable name="ALLOCATE(X,B045)" index="250644" value="0"/>
<variable name="ALLOCATE(X,B046)" index="250645" value="0"/>
<variable name="ALLOCATE(X,B047)" index="250646" value="0"/>
<variable name="ALLOCATE(X,B048)" index="250647" value="0"/>
<variable name="ALLOCATE(X,B049)" index="250648" value="0"/>
<variable name="ALLOCATE(X,B050)" index="250649" value="0"/>
<variable name="ALLOCATE(X,B051)" index="250650" value="0"/>
<variable name="ALLOCATE(X,B052)" index="250651" value="0"/>
<variable name="ALLOCATE(X,B053)" index="250652" value="0"/>
<variable name="ALLOCATE(X,B054)" index="250653" value="0"/>
<variable name="ALLOCATE(X,B055)" index="250654" value="0"/>
<variable name="ALLOCATE(X,B056)" index="250655" value="0"/>
<variable name="ALLOCATE(X,B057)" index="250656" value="0"/>
<variable name="ALLOCATE(X,B058)" index="250657" value="0"/>
<variable name="ALLOCATE(X,B059)" index="250658" value="0"/>
<variable name="ALLOCATE(X,B060)" index="250659" value="0"/>
<variable name="ALLOCATE(X,B061)" index="250660" value="1"/>
<variable name="ALLOCATE(X,B062)" index="250661" value="0"/>
<variable name="ALLOCATE(X,B063)" index="250662" value="0"/>
<variable name="ALLOCATE(X,B064)" index="250663" value="0"/>
<variable name="ALLOCATE(X,B065)" index="250664" value="0"/>
<variable name="ALLOCATE(X,B066)" index="250665" value="0"/>

```
<variable name="ALLOCATE(X,B067)" index="250666" value="0"/>
<variable name="ALLOCATE(X,B068)" index="250667" value="0"/>
<variable name="ALLOCATE(X,B069)" index="250668" value="0"/>
<variable name="ALLOCATE(X,B070)" index="250669" value="0"/>
<variable name="ALLOCATE(X,B071)" index="250670" value="0"/>
<variable name="ALLOCATE(X,B072)" index="250671" value="0"/>
<variable name="ALLOCATE(X,B073)" index="250672" value="0"/>
<variable name="ALLOCATE(X,B074)" index="250673" value="0"/>
<variable name="ALLOCATE(X,B075)" index="250674" value="0"/>
<variable name="ALLOCATE(X,B076)" index="250675" value="1"/>
<variable name="ALLOCATE(X,B077)" index="250676" value="0"/>
<variable name="ALLOCATE(X,B078)" index="250677" value="0"/>
<variable name="ALLOCATE(X,B079)" index="250678" value="0"/>
<variable name="ALLOCATE(X,B080)" index="250679" value="0"/>
<variable name="ALLOCATE(X,B081)" index="250680" value="0"/>
<variable name="ALLOCATE(X,B082)" index="250681" value="0"/>
<variable name="ALLOCATE(X,B083)" index="250682" value="0"/>
<variable name="ALLOCATE(X,B084)" index="250683" value="0"/>
<variable name="ALLOCATE(X,B085)" index="250684" value="0"/>
<variable name="ALLOCATE(X,B086)" index="250685" value="0"/>
<variable name="ALLOCATE(X,B087)" index="250686" value="0"/>
<variable name="ALLOCATE(X,B088)" index="250687" value="0"/>
<variable name="ALLOCATE(X,B089)" index="250688" value="0"/>
<variable name="ALLOCATE(X,B090)" index="250689" value="0"/>
<variable name="ALLOCATE(X,B091)" index="250690" value="0"/>
<variable name="ALLOCATE(X,B092)" index="250691" value="0"/>
<variable name="ALLOCATE(X,B093)" index="250692" value="0"/>
<variable name="ALLOCATE(X,B094)" index="250693" value="0"/>
<variable name="ALLOCATE(X,B095)" index="250694" value="0"/>
<variable name="ALLOCATE(X,B096)" index="250695" value="0"/>
<variable name="ALLOCATE(X,B097)" index="250696" value="0"/>
<variable name="ALLOCATE(X,B098)" index="250697" value="0"/>
<variable name="ALLOCATE(X,B099)" index="250698" value="0"/>
<variable name="ALLOCATE(X,B100)" index="250699" value="0"/>
<variable name="ALLOCATE(X,C001)" index="250700" value="0"/>
<variable name="ALLOCATE(X,C002)" index="250701" value="0"/>
<variable name="ALLOCATE(X,C003)" index="250702" value="0"/>
<variable name="ALLOCATE(X,C004)" index="250703" value="0"/>
<variable name="ALLOCATE(X,C005)" index="250704" value="0"/>
<variable name="ALLOCATE(X,C006)" index="250705" value="0"/>
<variable name="ALLOCATE(X,C007)" index="250706" value="0"/>
<variable name="ALLOCATE(X,C008)" index="250707" value="0"/>
<variable name="ALLOCATE(X,C009)" index="250708" value="0"/>
<variable name="ALLOCATE(X,C010)" index="250709" value="0"/>
<variable name="ALLOCATE(X,C011)" index="250710" value="0"/>
<variable name="ALLOCATE(X,C012)" index="250711" value="0"/>
<variable name="ALLOCATE(X,C013)" index="250712" value="0"/>
<variable name="ALLOCATE(X,C014)" index="250713" value="0"/>
<variable name="ALLOCATE(X,C015)" index="250714" value="0"/>
<variable name="ALLOCATE(X,C016)" index="250715" value="0"/>
<variable name="ALLOCATE(X,C017)" index="250716" value="0"/>
<variable name="ALLOCATE(X,C018)" index="250717" value="0"/>
<variable name="ALLOCATE(X,C019)" index="250718" value="0"/>
<variable name="ALLOCATE(X,C020)" index="250719" value="0"/>
<variable name="ALLOCATE(X,C021)" index="250720" value="0"/>
<variable name="ALLOCATE(X,C022)" index="250721" value="0"/>
<variable name="ALLOCATE(X,C023)" index="250722" value="0"/>
<variable name="ALLOCATE(X,C024)" index="250723" value="0"/>
<variable name="ALLOCATE(X,C025)" index="250724" value="0"/>
<variable name="ALLOCATE(X,C026)" index="250725" value="0"/>
<variable name="ALLOCATE(X,C027)" index="250726" value="0"/>
<variable name="ALLOCATE(X,C028)" index="250727" value="0"/>
<variable name="ALLOCATE(X,C029)" index="250728" value="0"/>
<variable name="ALLOCATE(X,C030)" index="250729" value="0"/>
<variable name="ALLOCATE(X,C031)" index="250730" value="0"/>
<variable name="ALLOCATE(X,C032)" index="250731" value="1"/>
<variable name="ALLOCATE(X,C033)" index="250732" value="0"/>
<variable name="ALLOCATE(X,C034)" index="250733" value="0"/>
<variable name="ALLOCATE(X,C035)" index="250734" value="0"/>
<variable name="ALLOCATE(X,C036)" index="250735" value="0"/>
<variable name="ALLOCATE(X,C037)" index="250736" value="0"/>
<variable name="ALLOCATE(X,C038)" index="250737" value="0"/>
```

<variable name="ALLOCATE(X,C039)" index="250738" value="0"/>
<variable name="ALLOCATE(X,C040)" index="250739" value="0"/>
<variable name="ALLOCATE(X,C041)" index="250740" value="0"/>
<variable name="ALLOCATE(X,C042)" index="250741" value="0"/>
<variable name="ALLOCATE(X,C043)" index="250742" value="0"/>
<variable name="ALLOCATE(X,C044)" index="250743" value="0"/>
<variable name="ALLOCATE(X,C045)" index="250744" value="0"/>
<variable name="ALLOCATE(X,C046)" index="250745" value="0"/>
<variable name="ALLOCATE(X,C047)" index="250746" value="0"/>
<variable name="ALLOCATE(X,C048)" index="250747" value="0"/>
<variable name="ALLOCATE(X,C049)" index="250748" value="0"/>
<variable name="ALLOCATE(X,C050)" index="250749" value="0"/>
<variable name="ALLOCATE(X,C051)" index="250750" value="1"/>
<variable name="ALLOCATE(X,C052)" index="250751" value="0"/>
<variable name="ALLOCATE(X,C053)" index="250752" value="0"/>
<variable name="ALLOCATE(X,C054)" index="250753" value="0"/>
<variable name="ALLOCATE(X,C055)" index="250754" value="0"/>
<variable name="ALLOCATE(X,C056)" index="250755" value="0"/>
<variable name="ALLOCATE(X,C057)" index="250756" value="0"/>
<variable name="ALLOCATE(X,C058)" index="250757" value="0"/>
<variable name="ALLOCATE(X,C059)" index="250758" value="0"/>
<variable name="ALLOCATE(X,C060)" index="250759" value="0"/>
<variable name="ALLOCATE(X,C061)" index="250760" value="0"/>
<variable name="ALLOCATE(X,C062)" index="250761" value="1"/>
<variable name="ALLOCATE(X,C063)" index="250762" value="0"/>
<variable name="ALLOCATE(X,C064)" index="250763" value="0"/>
<variable name="ALLOCATE(X,C065)" index="250764" value="0"/>
<variable name="ALLOCATE(X,C066)" index="250765" value="0"/>
<variable name="ALLOCATE(X,C067)" index="250766" value="0"/>
<variable name="ALLOCATE(X,C068)" index="250767" value="0"/>
<variable name="ALLOCATE(X,C069)" index="250768" value="0"/>
<variable name="ALLOCATE(X,C070)" index="250769" value="0"/>
<variable name="ALLOCATE(X,C071)" index="250770" value="0"/>
<variable name="ALLOCATE(X,C072)" index="250771" value="0"/>
<variable name="ALLOCATE(X,C073)" index="250772" value="0"/>
<variable name="ALLOCATE(X,C074)" index="250773" value="0"/>
<variable name="ALLOCATE(X,C075)" index="250774" value="0"/>
<variable name="ALLOCATE(X,C076)" index="250775" value="0"/>
<variable name="ALLOCATE(X,C077)" index="250776" value="1"/>
<variable name="ALLOCATE(X,C078)" index="250777" value="0"/>
<variable name="ALLOCATE(X,C079)" index="250778" value="0"/>
<variable name="ALLOCATE(X,C080)" index="250779" value="0"/>
<variable name="ALLOCATE(X,C081)" index="250780" value="0"/>
<variable name="ALLOCATE(X,C082)" index="250781" value="0"/>
<variable name="ALLOCATE(X,C083)" index="250782" value="0"/>
<variable name="ALLOCATE(X,C084)" index="250783" value="0"/>
<variable name="ALLOCATE(X,C085)" index="250784" value="0"/>
<variable name="ALLOCATE(X,C086)" index="250785" value="0"/>
<variable name="ALLOCATE(X,C087)" index="250786" value="0"/>
<variable name="ALLOCATE(X,C088)" index="250787" value="0"/>
<variable name="ALLOCATE(X,C089)" index="250788" value="0"/>
<variable name="ALLOCATE(X,C090)" index="250789" value="0"/>
<variable name="ALLOCATE(X,C091)" index="250790" value="0"/>
<variable name="ALLOCATE(X,C092)" index="250791" value="0"/>
<variable name="ALLOCATE(X,C093)" index="250792" value="0"/>
<variable name="ALLOCATE(X,C094)" index="250793" value="0"/>
<variable name="ALLOCATE(X,C095)" index="250794" value="0"/>
<variable name="ALLOCATE(X,C096)" index="250795" value="0"/>
<variable name="ALLOCATE(X,C097)" index="250796" value="0"/>
<variable name="ALLOCATE(X,C098)" index="250797" value="0"/>
<variable name="ALLOCATE(X,C099)" index="250798" value="0"/>
<variable name="ALLOCATE(X,C100)" index="250799" value="0"/>
<variable name="ALLOCATE(X,D001)" index="250800" value="0"/>
<variable name="ALLOCATE(X,D002)" index="250801" value="0"/>
<variable name="ALLOCATE(X,D003)" index="250802" value="1"/>
<variable name="ALLOCATE(X,D004)" index="250803" value="0"/>
<variable name="ALLOCATE(X,D005)" index="250804" value="0"/>
<variable name="ALLOCATE(X,D006)" index="250805" value="0"/>
<variable name="ALLOCATE(X,D007)" index="250806" value="0"/>
<variable name="ALLOCATE(X,D008)" index="250807" value="0"/>
<variable name="ALLOCATE(X,D009)" index="250808" value="0"/>
<variable name="ALLOCATE(X,D010)" index="250809" value="0"/>

<variable name="ALLOCATE(X,D011)" index="250810" value="0"/>
<variable name="ALLOCATE(X,D012)" index="250811" value="0"/>
<variable name="ALLOCATE(X,D013)" index="250812" value="0"/>
<variable name="ALLOCATE(X,D014)" index="250813" value="0"/>
<variable name="ALLOCATE(X,D015)" index="250814" value="0"/>
<variable name="ALLOCATE(X,D016)" index="250815" value="0"/>
<variable name="ALLOCATE(X,D017)" index="250816" value="0"/>
<variable name="ALLOCATE(X,D018)" index="250817" value="0"/>
<variable name="ALLOCATE(X,D019)" index="250818" value="0"/>
<variable name="ALLOCATE(X,D020)" index="250819" value="0"/>
<variable name="ALLOCATE(X,D021)" index="250820" value="0"/>
<variable name="ALLOCATE(X,D022)" index="250821" value="0"/>
<variable name="ALLOCATE(X,D023)" index="250822" value="1"/>
<variable name="ALLOCATE(X,D024)" index="250823" value="0"/>
<variable name="ALLOCATE(X,D025)" index="250824" value="0"/>
<variable name="ALLOCATE(X,D026)" index="250825" value="0"/>
<variable name="ALLOCATE(X,D027)" index="250826" value="0"/>
<variable name="ALLOCATE(X,D028)" index="250827" value="0"/>
<variable name="ALLOCATE(X,D029)" index="250828" value="0"/>
<variable name="ALLOCATE(X,D030)" index="250829" value="1"/>
<variable name="ALLOCATE(X,D031)" index="250830" value="0"/>
<variable name="ALLOCATE(X,D032)" index="250831" value="0"/>
<variable name="ALLOCATE(X,D033)" index="250832" value="0"/>
<variable name="ALLOCATE(X,D034)" index="250833" value="0"/>
<variable name="ALLOCATE(X,D035)" index="250834" value="0"/>
<variable name="ALLOCATE(X,D036)" index="250835" value="0"/>
<variable name="ALLOCATE(X,D037)" index="250836" value="0"/>
<variable name="ALLOCATE(X,D038)" index="250837" value="0"/>
<variable name="ALLOCATE(X,D039)" index="250838" value="0"/>
<variable name="ALLOCATE(X,D040)" index="250839" value="0"/>
<variable name="ALLOCATE(X,D041)" index="250840" value="0"/>
<variable name="ALLOCATE(X,D042)" index="250841" value="0"/>
<variable name="ALLOCATE(X,D043)" index="250842" value="0"/>
<variable name="ALLOCATE(X,D044)" index="250843" value="0"/>
<variable name="ALLOCATE(X,D045)" index="250844" value="0"/>
<variable name="ALLOCATE(X,D046)" index="250845" value="1"/>
<variable name="ALLOCATE(X,D047)" index="250846" value="0"/>
<variable name="ALLOCATE(X,D048)" index="250847" value="0"/>
<variable name="ALLOCATE(X,D049)" index="250848" value="0"/>
<variable name="ALLOCATE(X,D050)" index="250849" value="0"/>
<variable name="ALLOCATE(X,D051)" index="250850" value="0"/>
<variable name="ALLOCATE(X,D052)" index="250851" value="0"/>
<variable name="ALLOCATE(X,D053)" index="250852" value="1"/>
<variable name="ALLOCATE(X,D054)" index="250853" value="0"/>
<variable name="ALLOCATE(X,D055)" index="250854" value="0"/>
<variable name="ALLOCATE(X,D056)" index="250855" value="0"/>
<variable name="ALLOCATE(X,D057)" index="250856" value="0"/>
<variable name="ALLOCATE(X,D058)" index="250857" value="0"/>
<variable name="ALLOCATE(X,D059)" index="250858" value="0"/>
<variable name="ALLOCATE(X,D060)" index="250859" value="0"/>
<variable name="ALLOCATE(X,D061)" index="250860" value="0"/>
<variable name="ALLOCATE(X,D062)" index="250861" value="0"/>
<variable name="ALLOCATE(X,D063)" index="250862" value="0"/>
<variable name="ALLOCATE(X,D064)" index="250863" value="0"/>
<variable name="ALLOCATE(X,D065)" index="250864" value="0"/>
<variable name="ALLOCATE(X,D066)" index="250865" value="0"/>
<variable name="ALLOCATE(X,D067)" index="250866" value="0"/>
<variable name="ALLOCATE(X,D068)" index="250867" value="0"/>
<variable name="ALLOCATE(X,D069)" index="250868" value="0"/>
<variable name="ALLOCATE(X,D070)" index="250869" value="0"/>
<variable name="ALLOCATE(X,D071)" index="250870" value="0"/>
<variable name="ALLOCATE(X,D072)" index="250871" value="0"/>
<variable name="ALLOCATE(X,D073)" index="250872" value="0"/>
<variable name="ALLOCATE(X,D074)" index="250873" value="0"/>
<variable name="ALLOCATE(X,D075)" index="250874" value="0"/>
<variable name="ALLOCATE(X,D076)" index="250875" value="0"/>
<variable name="ALLOCATE(X,D077)" index="250876" value="0"/>
<variable name="ALLOCATE(X,D078)" index="250877" value="0"/>
<variable name="ALLOCATE(X,D079)" index="250878" value="0"/>
<variable name="ALLOCATE(X,D080)" index="250879" value="0"/>
<variable name="ALLOCATE(X,D081)" index="250880" value="0"/>
<variable name="ALLOCATE(X,D082)" index="250881" value="0"/>

```
<variable name="ALLOCATE(X,D083)" index="250882" value="0"/>
<variable name="ALLOCATE(X,D084)" index="250883" value="0"/>
<variable name="ALLOCATE(X,D085)" index="250884" value="0"/>
<variable name="ALLOCATE(X,D086)" index="250885" value="0"/>
<variable name="ALLOCATE(X,D087)" index="250886" value="0"/>
<variable name="ALLOCATE(X,D088)" index="250887" value="0"/>
<variable name="ALLOCATE(X,D089)" index="250888" value="0"/>
<variable name="ALLOCATE(X,D090)" index="250889" value="0"/>
<variable name="ALLOCATE(X,D091)" index="250890" value="0"/>
<variable name="ALLOCATE(X,D092)" index="250891" value="0"/>
<variable name="ALLOCATE(X,D093)" index="250892" value="0"/>
<variable name="ALLOCATE(X,D094)" index="250893" value="0"/>
<variable name="ALLOCATE(X,D095)" index="250894" value="0"/>
<variable name="ALLOCATE(X,D096)" index="250895" value="0"/>
<variable name="ALLOCATE(X,D097)" index="250896" value="0"/>
<variable name="ALLOCATE(X,D098)" index="250897" value="0"/>
<variable name="ALLOCATE(X,D099)" index="250898" value="0"/>
<variable name="ALLOCATE(X,D100)" index="250899" value="0"/>
<variable name="ALLOCATE(X,E001)" index="250900" value="0"/>
<variable name="ALLOCATE(X,E002)" index="250901" value="0"/>
<variable name="ALLOCATE(X,E003)" index="250902" value="0"/>
<variable name="ALLOCATE(X,E004)" index="250903" value="0"/>
<variable name="ALLOCATE(X,E005)" index="250904" value="0"/>
<variable name="ALLOCATE(X,E006)" index="250905" value="0"/>
<variable name="ALLOCATE(X,E007)" index="250906" value="0"/>
<variable name="ALLOCATE(X,E008)" index="250907" value="0"/>
<variable name="ALLOCATE(X,E009)" index="250908" value="0"/>
<variable name="ALLOCATE(X,E010)" index="250909" value="0"/>
<variable name="ALLOCATE(X,E011)" index="250910" value="0"/>
<variable name="ALLOCATE(X,E012)" index="250911" value="0"/>
<variable name="ALLOCATE(X,E013)" index="250912" value="0"/>
<variable name="ALLOCATE(X,E014)" index="250913" value="0"/>
<variable name="ALLOCATE(X,E015)" index="250914" value="1"/>
<variable name="ALLOCATE(X,E016)" index="250915" value="0"/>
<variable name="ALLOCATE(X,E017)" index="250916" value="0"/>
<variable name="ALLOCATE(X,E018)" index="250917" value="0"/>
<variable name="ALLOCATE(X,E019)" index="250918" value="0"/>
<variable name="ALLOCATE(X,E020)" index="250919" value="0"/>
<variable name="ALLOCATE(X,E021)" index="250920" value="0"/>
<variable name="ALLOCATE(X,E022)" index="250921" value="0"/>
<variable name="ALLOCATE(X,E023)" index="250922" value="0"/>
<variable name="ALLOCATE(X,E024)" index="250923" value="0"/>
<variable name="ALLOCATE(X,E025)" index="250924" value="0"/>
<variable name="ALLOCATE(X,E026)" index="250925" value="0"/>
<variable name="ALLOCATE(X,E027)" index="250926" value="0"/>
<variable name="ALLOCATE(X,E028)" index="250927" value="0"/>
<variable name="ALLOCATE(X,E029)" index="250928" value="0"/>
<variable name="ALLOCATE(X,E030)" index="250929" value="0"/>
<variable name="ALLOCATE(X,E031)" index="250930" value="0"/>
<variable name="ALLOCATE(X,E032)" index="250931" value="0"/>
<variable name="ALLOCATE(X,E033)" index="250932" value="0"/>
<variable name="ALLOCATE(X,E034)" index="250933" value="0"/>
<variable name="ALLOCATE(X,E035)" index="250934" value="0"/>
<variable name="ALLOCATE(X,E036)" index="250935" value="0"/>
<variable name="ALLOCATE(X,E037)" index="250936" value="0"/>
<variable name="ALLOCATE(X,E038)" index="250937" value="0"/>
<variable name="ALLOCATE(X,E039)" index="250938" value="0"/>
<variable name="ALLOCATE(X,E040)" index="250939" value="0"/>
<variable name="ALLOCATE(X,E041)" index="250940" value="0"/>
<variable name="ALLOCATE(X,E042)" index="250941" value="0"/>
<variable name="ALLOCATE(X,E043)" index="250942" value="0"/>
<variable name="ALLOCATE(X,E044)" index="250943" value="0"/>
<variable name="ALLOCATE(X,E045)" index="250944" value="0"/>
<variable name="ALLOCATE(X,E046)" index="250945" value="0"/>
<variable name="ALLOCATE(X,E047)" index="250946" value="0"/>
<variable name="ALLOCATE(X,E048)" index="250947" value="0"/>
<variable name="ALLOCATE(X,E049)" index="250948" value="1"/>
<variable name="ALLOCATE(X,E050)" index="250949" value="0"/>
<variable name="ALLOCATE(X,E051)" index="250950" value="0"/>
<variable name="ALLOCATE(X,E052)" index="250951" value="0"/>
<variable name="ALLOCATE(X,E053)" index="250952" value="0"/>
<variable name="ALLOCATE(X,E054)" index="250953" value="0"/>
```

<variable name="ALLOCATE(X,E055)" index="250954" value="0"/>
<variable name="ALLOCATE(X,E056)" index="250955" value="0"/>
<variable name="ALLOCATE(X,E057)" index="250956" value="0"/>
<variable name="ALLOCATE(X,E058)" index="250957" value="0"/>
<variable name="ALLOCATE(X,E059)" index="250958" value="0"/>
<variable name="ALLOCATE(X,E060)" index="250959" value="0"/>
<variable name="ALLOCATE(X,E061)" index="250960" value="0"/>
<variable name="ALLOCATE(X,E062)" index="250961" value="0"/>
<variable name="ALLOCATE(X,E063)" index="250962" value="0"/>
<variable name="ALLOCATE(X,E064)" index="250963" value="0"/>
<variable name="ALLOCATE(X,E065)" index="250964" value="0"/>
<variable name="ALLOCATE(X,E066)" index="250965" value="0"/>
<variable name="ALLOCATE(X,E067)" index="250966" value="0"/>
<variable name="ALLOCATE(X,E068)" index="250967" value="0"/>
<variable name="ALLOCATE(X,E069)" index="250968" value="0"/>
<variable name="ALLOCATE(X,E070)" index="250969" value="0"/>
<variable name="ALLOCATE(X,E071)" index="250970" value="0"/>
<variable name="ALLOCATE(X,E072)" index="250971" value="0"/>
<variable name="ALLOCATE(X,E073)" index="250972" value="0"/>
<variable name="ALLOCATE(X,E074)" index="250973" value="0"/>
<variable name="ALLOCATE(X,E075)" index="250974" value="0"/>
<variable name="ALLOCATE(X,E076)" index="250975" value="0"/>
<variable name="ALLOCATE(X,E077)" index="250976" value="0"/>
<variable name="ALLOCATE(X,E078)" index="250977" value="1"/>
<variable name="ALLOCATE(X,E079)" index="250978" value="0"/>
<variable name="ALLOCATE(X,E080)" index="250979" value="0"/>
<variable name="ALLOCATE(X,E081)" index="250980" value="0"/>
<variable name="ALLOCATE(X,E082)" index="250981" value="0"/>
<variable name="ALLOCATE(X,E083)" index="250982" value="0"/>
<variable name="ALLOCATE(X,E084)" index="250983" value="0"/>
<variable name="ALLOCATE(X,E085)" index="250984" value="0"/>
<variable name="ALLOCATE(X,E086)" index="250985" value="0"/>
<variable name="ALLOCATE(X,E087)" index="250986" value="0"/>
<variable name="ALLOCATE(X,E088)" index="250987" value="0"/>
<variable name="ALLOCATE(X,E089)" index="250988" value="0"/>
<variable name="ALLOCATE(X,E090)" index="250989" value="0"/>
<variable name="ALLOCATE(X,E091)" index="250990" value="0"/>
<variable name="ALLOCATE(X,E092)" index="250991" value="0"/>
<variable name="ALLOCATE(X,E093)" index="250992" value="0"/>
<variable name="ALLOCATE(X,E094)" index="250993" value="0"/>
<variable name="ALLOCATE(X,E095)" index="250994" value="0"/>
<variable name="ALLOCATE(X,E096)" index="250995" value="0"/>
<variable name="ALLOCATE(X,E097)" index="250996" value="0"/>
<variable name="ALLOCATE(X,E098)" index="250997" value="1"/>
<variable name="ALLOCATE(X,E099)" index="250998" value="0"/>
<variable name="ALLOCATE(X,E100)" index="250999" value="0"/>
<variable name="ALLOCATE(Y,A001)" index="251000" value="0"/>
<variable name="ALLOCATE(Y,A002)" index="251001" value="1"/>
<variable name="ALLOCATE(Y,A003)" index="251002" value="0"/>
<variable name="ALLOCATE(Y,A004)" index="251003" value="0"/>
<variable name="ALLOCATE(Y,A005)" index="251004" value="0"/>
<variable name="ALLOCATE(Y,A006)" index="251005" value="0"/>
<variable name="ALLOCATE(Y,A007)" index="251006" value="0"/>
<variable name="ALLOCATE(Y,A008)" index="251007" value="0"/>
<variable name="ALLOCATE(Y,A009)" index="251008" value="0"/>
<variable name="ALLOCATE(Y,A010)" index="251009" value="0"/>
<variable name="ALLOCATE(Y,A011)" index="251010" value="0"/>
<variable name="ALLOCATE(Y,A012)" index="251011" value="0"/>
<variable name="ALLOCATE(Y,A013)" index="251012" value="0"/>
<variable name="ALLOCATE(Y,A014)" index="251013" value="0"/>
<variable name="ALLOCATE(Y,A015)" index="251014" value="0"/>
<variable name="ALLOCATE(Y,A016)" index="251015" value="0"/>
<variable name="ALLOCATE(Y,A017)" index="251016" value="0"/>
<variable name="ALLOCATE(Y,A018)" index="251017" value="1"/>
<variable name="ALLOCATE(Y,A019)" index="251018" value="0"/>
<variable name="ALLOCATE(Y,A020)" index="251019" value="0"/>
<variable name="ALLOCATE(Y,A021)" index="251020" value="0"/>
<variable name="ALLOCATE(Y,A022)" index="251021" value="0"/>
<variable name="ALLOCATE(Y,A023)" index="251022" value="1"/>
<variable name="ALLOCATE(Y,A024)" index="251023" value="0"/>
<variable name="ALLOCATE(Y,A025)" index="251024" value="0"/>
<variable name="ALLOCATE(Y,A026)" index="251025" value="0"/>

<variable name="ALLOCATE(Y,A027)" index="251026" value="0"/>
<variable name="ALLOCATE(Y,A028)" index="251027" value="0"/>
<variable name="ALLOCATE(Y,A029)" index="251028" value="1"/>
<variable name="ALLOCATE(Y,A030)" index="251029" value="0"/>
<variable name="ALLOCATE(Y,A031)" index="251030" value="0"/>
<variable name="ALLOCATE(Y,A032)" index="251031" value="0"/>
<variable name="ALLOCATE(Y,A033)" index="251032" value="0"/>
<variable name="ALLOCATE(Y,A034)" index="251033" value="0"/>
<variable name="ALLOCATE(Y,A035)" index="251034" value="0"/>
<variable name="ALLOCATE(Y,A036)" index="251035" value="0"/>
<variable name="ALLOCATE(Y,A037)" index="251036" value="0"/>
<variable name="ALLOCATE(Y,A038)" index="251037" value="1"/>
<variable name="ALLOCATE(Y,A039)" index="251038" value="0"/>
<variable name="ALLOCATE(Y,A040)" index="251039" value="0"/>
<variable name="ALLOCATE(Y,A041)" index="251040" value="0"/>
<variable name="ALLOCATE(Y,A042)" index="251041" value="0"/>
<variable name="ALLOCATE(Y,A043)" index="251042" value="0"/>
<variable name="ALLOCATE(Y,A044)" index="251043" value="0"/>
<variable name="ALLOCATE(Y,A045)" index="251044" value="0"/>
<variable name="ALLOCATE(Y,A046)" index="251045" value="0"/>
<variable name="ALLOCATE(Y,A047)" index="251046" value="0"/>
<variable name="ALLOCATE(Y,A048)" index="251047" value="0"/>
<variable name="ALLOCATE(Y,A049)" index="251048" value="0"/>
<variable name="ALLOCATE(Y,A050)" index="251049" value="0"/>
<variable name="ALLOCATE(Y,A051)" index="251050" value="0"/>
<variable name="ALLOCATE(Y,A052)" index="251051" value="0"/>
<variable name="ALLOCATE(Y,A053)" index="251052" value="1"/>
<variable name="ALLOCATE(Y,A054)" index="251053" value="0"/>
<variable name="ALLOCATE(Y,A055)" index="251054" value="0"/>
<variable name="ALLOCATE(Y,A056)" index="251055" value="0"/>
<variable name="ALLOCATE(Y,A057)" index="251056" value="0"/>
<variable name="ALLOCATE(Y,A058)" index="251057" value="0"/>
<variable name="ALLOCATE(Y,A059)" index="251058" value="0"/>
<variable name="ALLOCATE(Y,A060)" index="251059" value="0"/>
<variable name="ALLOCATE(Y,A061)" index="251060" value="0"/>
<variable name="ALLOCATE(Y,A062)" index="251061" value="0"/>
<variable name="ALLOCATE(Y,A063)" index="251062" value="0"/>
<variable name="ALLOCATE(Y,A064)" index="251063" value="0"/>
<variable name="ALLOCATE(Y,A065)" index="251064" value="0"/>
<variable name="ALLOCATE(Y,A066)" index="251065" value="0"/>
<variable name="ALLOCATE(Y,A067)" index="251066" value="0"/>
<variable name="ALLOCATE(Y,A068)" index="251067" value="0"/>
<variable name="ALLOCATE(Y,A069)" index="251068" value="0"/>
<variable name="ALLOCATE(Y,A070)" index="251069" value="0"/>
<variable name="ALLOCATE(Y,A071)" index="251070" value="0"/>
<variable name="ALLOCATE(Y,A072)" index="251071" value="0"/>
<variable name="ALLOCATE(Y,A073)" index="251072" value="0"/>
<variable name="ALLOCATE(Y,A074)" index="251073" value="0"/>
<variable name="ALLOCATE(Y,A075)" index="251074" value="0"/>
<variable name="ALLOCATE(Y,A076)" index="251075" value="0"/>
<variable name="ALLOCATE(Y,A077)" index="251076" value="0"/>
<variable name="ALLOCATE(Y,A078)" index="251077" value="0"/>
<variable name="ALLOCATE(Y,A079)" index="251078" value="0"/>
<variable name="ALLOCATE(Y,A080)" index="251079" value="1"/>
<variable name="ALLOCATE(Y,A081)" index="251080" value="0"/>
<variable name="ALLOCATE(Y,A082)" index="251081" value="0"/>
<variable name="ALLOCATE(Y,A083)" index="251082" value="0"/>
<variable name="ALLOCATE(Y,A084)" index="251083" value="0"/>
<variable name="ALLOCATE(Y,A085)" index="251084" value="0"/>
<variable name="ALLOCATE(Y,A086)" index="251085" value="0"/>
<variable name="ALLOCATE(Y,A087)" index="251086" value="0"/>
<variable name="ALLOCATE(Y,A088)" index="251087" value="0"/>
<variable name="ALLOCATE(Y,A089)" index="251088" value="0"/>
<variable name="ALLOCATE(Y,A090)" index="251089" value="0"/>
<variable name="ALLOCATE(Y,A091)" index="251090" value="0"/>
<variable name="ALLOCATE(Y,A092)" index="251091" value="0"/>
<variable name="ALLOCATE(Y,A093)" index="251092" value="1"/>
<variable name="ALLOCATE(Y,A094)" index="251093" value="0"/>
<variable name="ALLOCATE(Y,A095)" index="251094" value="1"/>
<variable name="ALLOCATE(Y,A096)" index="251095" value="0"/>
<variable name="ALLOCATE(Y,A097)" index="251096" value="0"/>
<variable name="ALLOCATE(Y,A098)" index="251097" value="0"/>

<variable name="ALLOCATE(Y,A099)" index="251098" value="0"/>
<variable name="ALLOCATE(Y,A100)" index="251099" value="0"/>
<variable name="ALLOCATE(Y,B001)" index="251100" value="0"/>
<variable name="ALLOCATE(Y,B002)" index="251101" value="0"/>
<variable name="ALLOCATE(Y,B003)" index="251102" value="0"/>
<variable name="ALLOCATE(Y,B004)" index="251103" value="0"/>
<variable name="ALLOCATE(Y,B005)" index="251104" value="0"/>
<variable name="ALLOCATE(Y,B006)" index="251105" value="0"/>
<variable name="ALLOCATE(Y,B007)" index="251106" value="0"/>
<variable name="ALLOCATE(Y,B008)" index="251107" value="0"/>
<variable name="ALLOCATE(Y,B009)" index="251108" value="0"/>
<variable name="ALLOCATE(Y,B010)" index="251109" value="0"/>
<variable name="ALLOCATE(Y,B011)" index="251110" value="0"/>
<variable name="ALLOCATE(Y,B012)" index="251111" value="0"/>
<variable name="ALLOCATE(Y,B013)" index="251112" value="0"/>
<variable name="ALLOCATE(Y,B014)" index="251113" value="0"/>
<variable name="ALLOCATE(Y,B015)" index="251114" value="0"/>
<variable name="ALLOCATE(Y,B016)" index="251115" value="0"/>
<variable name="ALLOCATE(Y,B017)" index="251116" value="0"/>
<variable name="ALLOCATE(Y,B018)" index="251117" value="0"/>
<variable name="ALLOCATE(Y,B019)" index="251118" value="0"/>
<variable name="ALLOCATE(Y,B020)" index="251119" value="0"/>
<variable name="ALLOCATE(Y,B021)" index="251120" value="0"/>
<variable name="ALLOCATE(Y,B022)" index="251121" value="0"/>
<variable name="ALLOCATE(Y,B023)" index="251122" value="0"/>
<variable name="ALLOCATE(Y,B024)" index="251123" value="0"/>
<variable name="ALLOCATE(Y,B025)" index="251124" value="0"/>
<variable name="ALLOCATE(Y,B026)" index="251125" value="0"/>
<variable name="ALLOCATE(Y,B027)" index="251126" value="0"/>
<variable name="ALLOCATE(Y,B028)" index="251127" value="0"/>
<variable name="ALLOCATE(Y,B029)" index="251128" value="0"/>
<variable name="ALLOCATE(Y,B030)" index="251129" value="0"/>
<variable name="ALLOCATE(Y,B031)" index="251130" value="1"/>
<variable name="ALLOCATE(Y,B032)" index="251131" value="0"/>
<variable name="ALLOCATE(Y,B033)" index="251132" value="0"/>
<variable name="ALLOCATE(Y,B034)" index="251133" value="0"/>
<variable name="ALLOCATE(Y,B035)" index="251134" value="0"/>
<variable name="ALLOCATE(Y,B036)" index="251135" value="0"/>
<variable name="ALLOCATE(Y,B037)" index="251136" value="0"/>
<variable name="ALLOCATE(Y,B038)" index="251137" value="0"/>
<variable name="ALLOCATE(Y,B039)" index="251138" value="0"/>
<variable name="ALLOCATE(Y,B040)" index="251139" value="0"/>
<variable name="ALLOCATE(Y,B041)" index="251140" value="0"/>
<variable name="ALLOCATE(Y,B042)" index="251141" value="0"/>
<variable name="ALLOCATE(Y,B043)" index="251142" value="0"/>
<variable name="ALLOCATE(Y,B044)" index="251143" value="0"/>
<variable name="ALLOCATE(Y,B045)" index="251144" value="0"/>
<variable name="ALLOCATE(Y,B046)" index="251145" value="0"/>
<variable name="ALLOCATE(Y,B047)" index="251146" value="0"/>
<variable name="ALLOCATE(Y,B048)" index="251147" value="0"/>
<variable name="ALLOCATE(Y,B049)" index="251148" value="0"/>
<variable name="ALLOCATE(Y,B050)" index="251149" value="0"/>
<variable name="ALLOCATE(Y,B051)" index="251150" value="0"/>
<variable name="ALLOCATE(Y,B052)" index="251151" value="0"/>
<variable name="ALLOCATE(Y,B053)" index="251152" value="0"/>
<variable name="ALLOCATE(Y,B054)" index="251153" value="0"/>
<variable name="ALLOCATE(Y,B055)" index="251154" value="0"/>
<variable name="ALLOCATE(Y,B056)" index="251155" value="1"/>
<variable name="ALLOCATE(Y,B057)" index="251156" value="0"/>
<variable name="ALLOCATE(Y,B058)" index="251157" value="0"/>
<variable name="ALLOCATE(Y,B059)" index="251158" value="0"/>
<variable name="ALLOCATE(Y,B060)" index="251159" value="0"/>
<variable name="ALLOCATE(Y,B061)" index="251160" value="0"/>
<variable name="ALLOCATE(Y,B062)" index="251161" value="0"/>
<variable name="ALLOCATE(Y,B063)" index="251162" value="0"/>
<variable name="ALLOCATE(Y,B064)" index="251163" value="0"/>
<variable name="ALLOCATE(Y,B065)" index="251164" value="0"/>
<variable name="ALLOCATE(Y,B066)" index="251165" value="0"/>
<variable name="ALLOCATE(Y,B067)" index="251166" value="0"/>
<variable name="ALLOCATE(Y,B068)" index="251167" value="0"/>
<variable name="ALLOCATE(Y,B069)" index="251168" value="1"/>
<variable name="ALLOCATE(Y,B070)" index="251169" value="0"/>

```
<variable name="ALLOCATE(Y,B071)" index="251170" value="0"/>
<variable name="ALLOCATE(Y,B072)" index="251171" value="0"/>
<variable name="ALLOCATE(Y,B073)" index="251172" value="0"/>
<variable name="ALLOCATE(Y,B074)" index="251173" value="0"/>
<variable name="ALLOCATE(Y,B075)" index="251174" value="0"/>
<variable name="ALLOCATE(Y,B076)" index="251175" value="0"/>
<variable name="ALLOCATE(Y,B077)" index="251176" value="0"/>
<variable name="ALLOCATE(Y,B078)" index="251177" value="0"/>
<variable name="ALLOCATE(Y,B079)" index="251178" value="0"/>
<variable name="ALLOCATE(Y,B080)" index="251179" value="1"/>
<variable name="ALLOCATE(Y,B081)" index="251180" value="0"/>
<variable name="ALLOCATE(Y,B082)" index="251181" value="0"/>
<variable name="ALLOCATE(Y,B083)" index="251182" value="0"/>
<variable name="ALLOCATE(Y,B084)" index="251183" value="0"/>
<variable name="ALLOCATE(Y,B085)" index="251184" value="0"/>
<variable name="ALLOCATE(Y,B086)" index="251185" value="0"/>
<variable name="ALLOCATE(Y,B087)" index="251186" value="0"/>
<variable name="ALLOCATE(Y,B088)" index="251187" value="0"/>
<variable name="ALLOCATE(Y,B089)" index="251188" value="1"/>
<variable name="ALLOCATE(Y,B090)" index="251189" value="0"/>
<variable name="ALLOCATE(Y,B091)" index="251190" value="0"/>
<variable name="ALLOCATE(Y,B092)" index="251191" value="0"/>
<variable name="ALLOCATE(Y,B093)" index="251192" value="0"/>
<variable name="ALLOCATE(Y,B094)" index="251193" value="0"/>
<variable name="ALLOCATE(Y,B095)" index="251194" value="0"/>
<variable name="ALLOCATE(Y,B096)" index="251195" value="0"/>
<variable name="ALLOCATE(Y,B097)" index="251196" value="1"/>
<variable name="ALLOCATE(Y,B098)" index="251197" value="0"/>
<variable name="ALLOCATE(Y,B099)" index="251198" value="0"/>
<variable name="ALLOCATE(Y,B100)" index="251199" value="0"/>
<variable name="ALLOCATE(Y,C001)" index="251200" value="0"/>
<variable name="ALLOCATE(Y,C002)" index="251201" value="0"/>
<variable name="ALLOCATE(Y,C003)" index="251202" value="0"/>
<variable name="ALLOCATE(Y,C004)" index="251203" value="0"/>
<variable name="ALLOCATE(Y,C005)" index="251204" value="0"/>
<variable name="ALLOCATE(Y,C006)" index="251205" value="0"/>
<variable name="ALLOCATE(Y,C007)" index="251206" value="0"/>
<variable name="ALLOCATE(Y,C008)" index="251207" value="0"/>
<variable name="ALLOCATE(Y,C009)" index="251208" value="0"/>
<variable name="ALLOCATE(Y,C010)" index="251209" value="0"/>
<variable name="ALLOCATE(Y,C011)" index="251210" value="0"/>
<variable name="ALLOCATE(Y,C012)" index="251211" value="0"/>
<variable name="ALLOCATE(Y,C013)" index="251212" value="0"/>
<variable name="ALLOCATE(Y,C014)" index="251213" value="0"/>
<variable name="ALLOCATE(Y,C015)" index="251214" value="0"/>
<variable name="ALLOCATE(Y,C016)" index="251215" value="0"/>
<variable name="ALLOCATE(Y,C017)" index="251216" value="0"/>
<variable name="ALLOCATE(Y,C018)" index="251217" value="0"/>
<variable name="ALLOCATE(Y,C019)" index="251218" value="0"/>
<variable name="ALLOCATE(Y,C020)" index="251219" value="0"/>
<variable name="ALLOCATE(Y,C021)" index="251220" value="0"/>
<variable name="ALLOCATE(Y,C022)" index="251221" value="0"/>
<variable name="ALLOCATE(Y,C023)" index="251222" value="0"/>
<variable name="ALLOCATE(Y,C024)" index="251223" value="0"/>
<variable name="ALLOCATE(Y,C025)" index="251224" value="0"/>
<variable name="ALLOCATE(Y,C026)" index="251225" value="0"/>
<variable name="ALLOCATE(Y,C027)" index="251226" value="0"/>
<variable name="ALLOCATE(Y,C028)" index="251227" value="0"/>
<variable name="ALLOCATE(Y,C029)" index="251228" value="0"/>
<variable name="ALLOCATE(Y,C030)" index="251229" value="0"/>
<variable name="ALLOCATE(Y,C031)" index="251230" value="0"/>
<variable name="ALLOCATE(Y,C032)" index="251231" value="0"/>
<variable name="ALLOCATE(Y,C033)" index="251232" value="0"/>
<variable name="ALLOCATE(Y,C034)" index="251233" value="0"/>
<variable name="ALLOCATE(Y,C035)" index="251234" value="0"/>
<variable name="ALLOCATE(Y,C036)" index="251235" value="0"/>
<variable name="ALLOCATE(Y,C037)" index="251236" value="0"/>
<variable name="ALLOCATE(Y,C038)" index="251237" value="0"/>
<variable name="ALLOCATE(Y,C039)" index="251238" value="0"/>
<variable name="ALLOCATE(Y,C040)" index="251239" value="0"/>
<variable name="ALLOCATE(Y,C041)" index="251240" value="0"/>
<variable name="ALLOCATE(Y,C042)" index="251241" value="0"/>
```

<variable name="ALLOCATE(Y,C043)" index="251242" value="0"/>
<variable name="ALLOCATE(Y,C044)" index="251243" value="0"/>
<variable name="ALLOCATE(Y,C045)" index="251244" value="0"/>
<variable name="ALLOCATE(Y,C046)" index="251245" value="0"/>
<variable name="ALLOCATE(Y,C047)" index="251246" value="0"/>
<variable name="ALLOCATE(Y,C048)" index="251247" value="0"/>
<variable name="ALLOCATE(Y,C049)" index="251248" value="0"/>
<variable name="ALLOCATE(Y,C050)" index="251249" value="0"/>
<variable name="ALLOCATE(Y,C051)" index="251250" value="0"/>
<variable name="ALLOCATE(Y,C052)" index="251251" value="0"/>
<variable name="ALLOCATE(Y,C053)" index="251252" value="0"/>
<variable name="ALLOCATE(Y,C054)" index="251253" value="0"/>
<variable name="ALLOCATE(Y,C055)" index="251254" value="0"/>
<variable name="ALLOCATE(Y,C056)" index="251255" value="0"/>
<variable name="ALLOCATE(Y,C057)" index="251256" value="0"/>
<variable name="ALLOCATE(Y,C058)" index="251257" value="1"/>
<variable name="ALLOCATE(Y,C059)" index="251258" value="0"/>
<variable name="ALLOCATE(Y,C060)" index="251259" value="0"/>
<variable name="ALLOCATE(Y,C061)" index="251260" value="1"/>
<variable name="ALLOCATE(Y,C062)" index="251261" value="0"/>
<variable name="ALLOCATE(Y,C063)" index="251262" value="0"/>
<variable name="ALLOCATE(Y,C064)" index="251263" value="0"/>
<variable name="ALLOCATE(Y,C065)" index="251264" value="0"/>
<variable name="ALLOCATE(Y,C066)" index="251265" value="0"/>
<variable name="ALLOCATE(Y,C067)" index="251266" value="0"/>
<variable name="ALLOCATE(Y,C068)" index="251267" value="0"/>
<variable name="ALLOCATE(Y,C069)" index="251268" value="0"/>
<variable name="ALLOCATE(Y,C070)" index="251269" value="0"/>
<variable name="ALLOCATE(Y,C071)" index="251270" value="0"/>
<variable name="ALLOCATE(Y,C072)" index="251271" value="0"/>
<variable name="ALLOCATE(Y,C073)" index="251272" value="0"/>
<variable name="ALLOCATE(Y,C074)" index="251273" value="0"/>
<variable name="ALLOCATE(Y,C075)" index="251274" value="0"/>
<variable name="ALLOCATE(Y,C076)" index="251275" value="0"/>
<variable name="ALLOCATE(Y,C077)" index="251276" value="0"/>
<variable name="ALLOCATE(Y,C078)" index="251277" value="0"/>
<variable name="ALLOCATE(Y,C079)" index="251278" value="0"/>
<variable name="ALLOCATE(Y,C080)" index="251279" value="1"/>
<variable name="ALLOCATE(Y,C081)" index="251280" value="0"/>
<variable name="ALLOCATE(Y,C082)" index="251281" value="0"/>
<variable name="ALLOCATE(Y,C083)" index="251282" value="0"/>
<variable name="ALLOCATE(Y,C084)" index="251283" value="0"/>
<variable name="ALLOCATE(Y,C085)" index="251284" value="0"/>
<variable name="ALLOCATE(Y,C086)" index="251285" value="0"/>
<variable name="ALLOCATE(Y,C087)" index="251286" value="0"/>
<variable name="ALLOCATE(Y,C088)" index="251287" value="0"/>
<variable name="ALLOCATE(Y,C089)" index="251288" value="0"/>
<variable name="ALLOCATE(Y,C090)" index="251289" value="0"/>
<variable name="ALLOCATE(Y,C091)" index="251290" value="0"/>
<variable name="ALLOCATE(Y,C092)" index="251291" value="0"/>
<variable name="ALLOCATE(Y,C093)" index="251292" value="0"/>
<variable name="ALLOCATE(Y,C094)" index="251293" value="0"/>
<variable name="ALLOCATE(Y,C095)" index="251294" value="0"/>
<variable name="ALLOCATE(Y,C096)" index="251295" value="0"/>
<variable name="ALLOCATE(Y,C097)" index="251296" value="0"/>
<variable name="ALLOCATE(Y,C098)" index="251297" value="0"/>
<variable name="ALLOCATE(Y,C099)" index="251298" value="0"/>
<variable name="ALLOCATE(Y,C100)" index="251299" value="0"/>
<variable name="ALLOCATE(Y,D001)" index="251300" value="0"/>
<variable name="ALLOCATE(Y,D002)" index="251301" value="0"/>
<variable name="ALLOCATE(Y,D003)" index="251302" value="0"/>
<variable name="ALLOCATE(Y,D004)" index="251303" value="0"/>
<variable name="ALLOCATE(Y,D005)" index="251304" value="1"/>
<variable name="ALLOCATE(Y,D006)" index="251305" value="0"/>
<variable name="ALLOCATE(Y,D007)" index="251306" value="0"/>
<variable name="ALLOCATE(Y,D008)" index="251307" value="0"/>
<variable name="ALLOCATE(Y,D009)" index="251308" value="0"/>
<variable name="ALLOCATE(Y,D010)" index="251309" value="0"/>
<variable name="ALLOCATE(Y,D011)" index="251310" value="0"/>
<variable name="ALLOCATE(Y,D012)" index="251311" value="0"/>
<variable name="ALLOCATE(Y,D013)" index="251312" value="0"/>
<variable name="ALLOCATE(Y,D014)" index="251313" value="0"/>

<variable name="ALLOCATE(Y,D015)" index="251314" value="0"/>
<variable name="ALLOCATE(Y,D016)" index="251315" value="0"/>
<variable name="ALLOCATE(Y,D017)" index="251316" value="0"/>
<variable name="ALLOCATE(Y,D018)" index="251317" value="0"/>
<variable name="ALLOCATE(Y,D019)" index="251318" value="0"/>
<variable name="ALLOCATE(Y,D020)" index="251319" value="1"/>
<variable name="ALLOCATE(Y,D021)" index="251320" value="0"/>
<variable name="ALLOCATE(Y,D022)" index="251321" value="0"/>
<variable name="ALLOCATE(Y,D023)" index="251322" value="0"/>
<variable name="ALLOCATE(Y,D024)" index="251323" value="0"/>
<variable name="ALLOCATE(Y,D025)" index="251324" value="1"/>
<variable name="ALLOCATE(Y,D026)" index="251325" value="0"/>
<variable name="ALLOCATE(Y,D027)" index="251326" value="1"/>
<variable name="ALLOCATE(Y,D028)" index="251327" value="0"/>
<variable name="ALLOCATE(Y,D029)" index="251328" value="0"/>
<variable name="ALLOCATE(Y,D030)" index="251329" value="0"/>
<variable name="ALLOCATE(Y,D031)" index="251330" value="0"/>
<variable name="ALLOCATE(Y,D032)" index="251331" value="0"/>
<variable name="ALLOCATE(Y,D033)" index="251332" value="0"/>
<variable name="ALLOCATE(Y,D034)" index="251333" value="0"/>
<variable name="ALLOCATE(Y,D035)" index="251334" value="0"/>
<variable name="ALLOCATE(Y,D036)" index="251335" value="0"/>
<variable name="ALLOCATE(Y,D037)" index="251336" value="0"/>
<variable name="ALLOCATE(Y,D038)" index="251337" value="0"/>
<variable name="ALLOCATE(Y,D039)" index="251338" value="1"/>
<variable name="ALLOCATE(Y,D040)" index="251339" value="0"/>
<variable name="ALLOCATE(Y,D041)" index="251340" value="0"/>
<variable name="ALLOCATE(Y,D042)" index="251341" value="0"/>
<variable name="ALLOCATE(Y,D043)" index="251342" value="0"/>
<variable name="ALLOCATE(Y,D044)" index="251343" value="0"/>
<variable name="ALLOCATE(Y,D045)" index="251344" value="0"/>
<variable name="ALLOCATE(Y,D046)" index="251345" value="0"/>
<variable name="ALLOCATE(Y,D047)" index="251346" value="0"/>
<variable name="ALLOCATE(Y,D048)" index="251347" value="0"/>
<variable name="ALLOCATE(Y,D049)" index="251348" value="0"/>
<variable name="ALLOCATE(Y,D050)" index="251349" value="1"/>
<variable name="ALLOCATE(Y,D051)" index="251350" value="0"/>
<variable name="ALLOCATE(Y,D052)" index="251351" value="0"/>
<variable name="ALLOCATE(Y,D053)" index="251352" value="0"/>
<variable name="ALLOCATE(Y,D054)" index="251353" value="0"/>
<variable name="ALLOCATE(Y,D055)" index="251354" value="0"/>
<variable name="ALLOCATE(Y,D056)" index="251355" value="1"/>
<variable name="ALLOCATE(Y,D057)" index="251356" value="0"/>
<variable name="ALLOCATE(Y,D058)" index="251357" value="0"/>
<variable name="ALLOCATE(Y,D059)" index="251358" value="0"/>
<variable name="ALLOCATE(Y,D060)" index="251359" value="0"/>
<variable name="ALLOCATE(Y,D061)" index="251360" value="1"/>
<variable name="ALLOCATE(Y,D062)" index="251361" value="0"/>
<variable name="ALLOCATE(Y,D063)" index="251362" value="0"/>
<variable name="ALLOCATE(Y,D064)" index="251363" value="0"/>
<variable name="ALLOCATE(Y,D065)" index="251364" value="0"/>
<variable name="ALLOCATE(Y,D066)" index="251365" value="0"/>
<variable name="ALLOCATE(Y,D067)" index="251366" value="0"/>
<variable name="ALLOCATE(Y,D068)" index="251367" value="0"/>
<variable name="ALLOCATE(Y,D069)" index="251368" value="0"/>
<variable name="ALLOCATE(Y,D070)" index="251369" value="0"/>
<variable name="ALLOCATE(Y,D071)" index="251370" value="0"/>
<variable name="ALLOCATE(Y,D072)" index="251371" value="0"/>
<variable name="ALLOCATE(Y,D073)" index="251372" value="0"/>
<variable name="ALLOCATE(Y,D074)" index="251373" value="0"/>
<variable name="ALLOCATE(Y,D075)" index="251374" value="0"/>
<variable name="ALLOCATE(Y,D076)" index="251375" value="0"/>
<variable name="ALLOCATE(Y,D077)" index="251376" value="0"/>
<variable name="ALLOCATE(Y,D078)" index="251377" value="0"/>
<variable name="ALLOCATE(Y,D079)" index="251378" value="0"/>
<variable name="ALLOCATE(Y,D080)" index="251379" value="0"/>
<variable name="ALLOCATE(Y,D081)" index="251380" value="0"/>
<variable name="ALLOCATE(Y,D082)" index="251381" value="0"/>
<variable name="ALLOCATE(Y,D083)" index="251382" value="0"/>
<variable name="ALLOCATE(Y,D084)" index="251383" value="0"/>
<variable name="ALLOCATE(Y,D085)" index="251384" value="0"/>
<variable name="ALLOCATE(Y,D086)" index="251385" value="0"/>

<variable name="ALLOCATE(Y,D087)" index="251386" value="0"/>
<variable name="ALLOCATE(Y,D088)" index="251387" value="0"/>
<variable name="ALLOCATE(Y,D089)" index="251388" value="1"/>
<variable name="ALLOCATE(Y,D090)" index="251389" value="0"/>
<variable name="ALLOCATE(Y,D091)" index="251390" value="0"/>
<variable name="ALLOCATE(Y,D092)" index="251391" value="0"/>
<variable name="ALLOCATE(Y,D093)" index="251392" value="0"/>
<variable name="ALLOCATE(Y,D094)" index="251393" value="0"/>
<variable name="ALLOCATE(Y,D095)" index="251394" value="0"/>
<variable name="ALLOCATE(Y,D096)" index="251395" value="0"/>
<variable name="ALLOCATE(Y,D097)" index="251396" value="0"/>
<variable name="ALLOCATE(Y,D098)" index="251397" value="0"/>
<variable name="ALLOCATE(Y,D099)" index="251398" value="0"/>
<variable name="ALLOCATE(Y,D100)" index="251399" value="1"/>
<variable name="ALLOCATE(Y,E001)" index="251400" value="0"/>
<variable name="ALLOCATE(Y,E002)" index="251401" value="0"/>
<variable name="ALLOCATE(Y,E003)" index="251402" value="0"/>
<variable name="ALLOCATE(Y,E004)" index="251403" value="0"/>
<variable name="ALLOCATE(Y,E005)" index="251404" value="0"/>
<variable name="ALLOCATE(Y,E006)" index="251405" value="0"/>
<variable name="ALLOCATE(Y,E007)" index="251406" value="0"/>
<variable name="ALLOCATE(Y,E008)" index="251407" value="0"/>
<variable name="ALLOCATE(Y,E009)" index="251408" value="0"/>
<variable name="ALLOCATE(Y,E010)" index="251409" value="0"/>
<variable name="ALLOCATE(Y,E011)" index="251410" value="0"/>
<variable name="ALLOCATE(Y,E012)" index="251411" value="0"/>
<variable name="ALLOCATE(Y,E013)" index="251412" value="0"/>
<variable name="ALLOCATE(Y,E014)" index="251413" value="0"/>
<variable name="ALLOCATE(Y,E015)" index="251414" value="0"/>
<variable name="ALLOCATE(Y,E016)" index="251415" value="0"/>
<variable name="ALLOCATE(Y,E017)" index="251416" value="0"/>
<variable name="ALLOCATE(Y,E018)" index="251417" value="0"/>
<variable name="ALLOCATE(Y,E019)" index="251418" value="0"/>
<variable name="ALLOCATE(Y,E020)" index="251419" value="0"/>
<variable name="ALLOCATE(Y,E021)" index="251420" value="0"/>
<variable name="ALLOCATE(Y,E022)" index="251421" value="0"/>
<variable name="ALLOCATE(Y,E023)" index="251422" value="0"/>
<variable name="ALLOCATE(Y,E024)" index="251423" value="0"/>
<variable name="ALLOCATE(Y,E025)" index="251424" value="0"/>
<variable name="ALLOCATE(Y,E026)" index="251425" value="1"/>
<variable name="ALLOCATE(Y,E027)" index="251426" value="0"/>
<variable name="ALLOCATE(Y,E028)" index="251427" value="0"/>
<variable name="ALLOCATE(Y,E029)" index="251428" value="0"/>
<variable name="ALLOCATE(Y,E030)" index="251429" value="1"/>
<variable name="ALLOCATE(Y,E031)" index="251430" value="0"/>
<variable name="ALLOCATE(Y,E032)" index="251431" value="0"/>
<variable name="ALLOCATE(Y,E033)" index="251432" value="0"/>
<variable name="ALLOCATE(Y,E034)" index="251433" value="0"/>
<variable name="ALLOCATE(Y,E035)" index="251434" value="0"/>
<variable name="ALLOCATE(Y,E036)" index="251435" value="0"/>
<variable name="ALLOCATE(Y,E037)" index="251436" value="0"/>
<variable name="ALLOCATE(Y,E038)" index="251437" value="0"/>
<variable name="ALLOCATE(Y,E039)" index="251438" value="0"/>
<variable name="ALLOCATE(Y,E040)" index="251439" value="0"/>
<variable name="ALLOCATE(Y,E041)" index="251440" value="0"/>
<variable name="ALLOCATE(Y,E042)" index="251441" value="0"/>
<variable name="ALLOCATE(Y,E043)" index="251442" value="0"/>
<variable name="ALLOCATE(Y,E044)" index="251443" value="0"/>
<variable name="ALLOCATE(Y,E045)" index="251444" value="0"/>
<variable name="ALLOCATE(Y,E046)" index="251445" value="0"/>
<variable name="ALLOCATE(Y,E047)" index="251446" value="0"/>
<variable name="ALLOCATE(Y,E048)" index="251447" value="0"/>
<variable name="ALLOCATE(Y,E049)" index="251448" value="0"/>
<variable name="ALLOCATE(Y,E050)" index="251449" value="0"/>
<variable name="ALLOCATE(Y,E051)" index="251450" value="0"/>
<variable name="ALLOCATE(Y,E052)" index="251451" value="0"/>
<variable name="ALLOCATE(Y,E053)" index="251452" value="0"/>
<variable name="ALLOCATE(Y,E054)" index="251453" value="1"/>
<variable name="ALLOCATE(Y,E055)" index="251454" value="1"/>
<variable name="ALLOCATE(Y,E056)" index="251455" value="0"/>
<variable name="ALLOCATE(Y,E057)" index="251456" value="0"/>
<variable name="ALLOCATE(Y,E058)" index="251457" value="0"/>

<variable name="ALLOCATE(Y,E059)" index="251458" value="0"/>
<variable name="ALLOCATE(Y,E060)" index="251459" value="0"/>
<variable name="ALLOCATE(Y,E061)" index="251460" value="0"/>
<variable name="ALLOCATE(Y,E062)" index="251461" value="0"/>
<variable name="ALLOCATE(Y,E063)" index="251462" value="0"/>
<variable name="ALLOCATE(Y,E064)" index="251463" value="0"/>
<variable name="ALLOCATE(Y,E065)" index="251464" value="0"/>
<variable name="ALLOCATE(Y,E066)" index="251465" value="0"/>
<variable name="ALLOCATE(Y,E067)" index="251466" value="0"/>
<variable name="ALLOCATE(Y,E068)" index="251467" value="0"/>
<variable name="ALLOCATE(Y,E069)" index="251468" value="0"/>
<variable name="ALLOCATE(Y,E070)" index="251469" value="0"/>
<variable name="ALLOCATE(Y,E071)" index="251470" value="1"/>
<variable name="ALLOCATE(Y,E072)" index="251471" value="0"/>
<variable name="ALLOCATE(Y,E073)" index="251472" value="0"/>
<variable name="ALLOCATE(Y,E074)" index="251473" value="0"/>
<variable name="ALLOCATE(Y,E075)" index="251474" value="0"/>
<variable name="ALLOCATE(Y,E076)" index="251475" value="0"/>
<variable name="ALLOCATE(Y,E077)" index="251476" value="0"/>
<variable name="ALLOCATE(Y,E078)" index="251477" value="0"/>
<variable name="ALLOCATE(Y,E079)" index="251478" value="0"/>
<variable name="ALLOCATE(Y,E080)" index="251479" value="0"/>
<variable name="ALLOCATE(Y,E081)" index="251480" value="0"/>
<variable name="ALLOCATE(Y,E082)" index="251481" value="1"/>
<variable name="ALLOCATE(Y,E083)" index="251482" value="1"/>
<variable name="ALLOCATE(Y,E084)" index="251483" value="0"/>
<variable name="ALLOCATE(Y,E085)" index="251484" value="0"/>
<variable name="ALLOCATE(Y,E086)" index="251485" value="0"/>
<variable name="ALLOCATE(Y,E087)" index="251486" value="0"/>
<variable name="ALLOCATE(Y,E088)" index="251487" value="0"/>
<variable name="ALLOCATE(Y,E089)" index="251488" value="1"/>
<variable name="ALLOCATE(Y,E090)" index="251489" value="0"/>
<variable name="ALLOCATE(Y,E091)" index="251490" value="0"/>
<variable name="ALLOCATE(Y,E092)" index="251491" value="0"/>
<variable name="ALLOCATE(Y,E093)" index="251492" value="0"/>
<variable name="ALLOCATE(Y,E094)" index="251493" value="0"/>
<variable name="ALLOCATE(Y,E095)" index="251494" value="0"/>
<variable name="ALLOCATE(Y,E096)" index="251495" value="0"/>
<variable name="ALLOCATE(Y,E097)" index="251496" value="0"/>
<variable name="ALLOCATE(Y,E098)" index="251497" value="0"/>
<variable name="ALLOCATE(Y,E099)" index="251498" value="1"/>
<variable name="ALLOCATE(Y,E100)" index="251499" value="0"/>
<variable name="ALLOCATE(Z,A001)" index="251500" value="0"/>
<variable name="ALLOCATE(Z,A002)" index="251501" value="0"/>
<variable name="ALLOCATE(Z,A003)" index="251502" value="0"/>
<variable name="ALLOCATE(Z,A004)" index="251503" value="0"/>
<variable name="ALLOCATE(Z,A005)" index="251504" value="1"/>
<variable name="ALLOCATE(Z,A006)" index="251505" value="0"/>
<variable name="ALLOCATE(Z,A007)" index="251506" value="0"/>
<variable name="ALLOCATE(Z,A008)" index="251507" value="0"/>
<variable name="ALLOCATE(Z,A009)" index="251508" value="0"/>
<variable name="ALLOCATE(Z,A010)" index="251509" value="0"/>
<variable name="ALLOCATE(Z,A011)" index="251510" value="0"/>
<variable name="ALLOCATE(Z,A012)" index="251511" value="0"/>
<variable name="ALLOCATE(Z,A013)" index="251512" value="0"/>
<variable name="ALLOCATE(Z,A014)" index="251513" value="0"/>
<variable name="ALLOCATE(Z,A015)" index="251514" value="0"/>
<variable name="ALLOCATE(Z,A016)" index="251515" value="0"/>
<variable name="ALLOCATE(Z,A017)" index="251516" value="0"/>
<variable name="ALLOCATE(Z,A018)" index="251517" value="0"/>
<variable name="ALLOCATE(Z,A019)" index="251518" value="0"/>
<variable name="ALLOCATE(Z,A020)" index="251519" value="0"/>
<variable name="ALLOCATE(Z,A021)" index="251520" value="0"/>
<variable name="ALLOCATE(Z,A022)" index="251521" value="0"/>
<variable name="ALLOCATE(Z,A023)" index="251522" value="0"/>
<variable name="ALLOCATE(Z,A024)" index="251523" value="0"/>
<variable name="ALLOCATE(Z,A025)" index="251524" value="0"/>
<variable name="ALLOCATE(Z,A026)" index="251525" value="0"/>
<variable name="ALLOCATE(Z,A027)" index="251526" value="0"/>
<variable name="ALLOCATE(Z,A028)" index="251527" value="0"/>
<variable name="ALLOCATE(Z,A029)" index="251528" value="0"/>
<variable name="ALLOCATE(Z,A030)" index="251529" value="0"/>

<variable name="ALLOCATE(Z,A031)" index="251530" value="0"/>
<variable name="ALLOCATE(Z,A032)" index="251531" value="0"/>
<variable name="ALLOCATE(Z,A033)" index="251532" value="0"/>
<variable name="ALLOCATE(Z,A034)" index="251533" value="0"/>
<variable name="ALLOCATE(Z,A035)" index="251534" value="0"/>
<variable name="ALLOCATE(Z,A036)" index="251535" value="0"/>
<variable name="ALLOCATE(Z,A037)" index="251536" value="1"/>
<variable name="ALLOCATE(Z,A038)" index="251537" value="0"/>
<variable name="ALLOCATE(Z,A039)" index="251538" value="0"/>
<variable name="ALLOCATE(Z,A040)" index="251539" value="0"/>
<variable name="ALLOCATE(Z,A041)" index="251540" value="0"/>
<variable name="ALLOCATE(Z,A042)" index="251541" value="0"/>
<variable name="ALLOCATE(Z,A043)" index="251542" value="0"/>
<variable name="ALLOCATE(Z,A044)" index="251543" value="0"/>
<variable name="ALLOCATE(Z,A045)" index="251544" value="0"/>
<variable name="ALLOCATE(Z,A046)" index="251545" value="0"/>
<variable name="ALLOCATE(Z,A047)" index="251546" value="1"/>
<variable name="ALLOCATE(Z,A048)" index="251547" value="0"/>
<variable name="ALLOCATE(Z,A049)" index="251548" value="0"/>
<variable name="ALLOCATE(Z,A050)" index="251549" value="0"/>
<variable name="ALLOCATE(Z,A051)" index="251550" value="0"/>
<variable name="ALLOCATE(Z,A052)" index="251551" value="0"/>
<variable name="ALLOCATE(Z,A053)" index="251552" value="0"/>
<variable name="ALLOCATE(Z,A054)" index="251553" value="0"/>
<variable name="ALLOCATE(Z,A055)" index="251554" value="0"/>
<variable name="ALLOCATE(Z,A056)" index="251555" value="0"/>
<variable name="ALLOCATE(Z,A057)" index="251556" value="0"/>
<variable name="ALLOCATE(Z,A058)" index="251557" value="0"/>
<variable name="ALLOCATE(Z,A059)" index="251558" value="0"/>
<variable name="ALLOCATE(Z,A060)" index="251559" value="0"/>
<variable name="ALLOCATE(Z,A061)" index="251560" value="0"/>
<variable name="ALLOCATE(Z,A062)" index="251561" value="0"/>
<variable name="ALLOCATE(Z,A063)" index="251562" value="0"/>
<variable name="ALLOCATE(Z,A064)" index="251563" value="0"/>
<variable name="ALLOCATE(Z,A065)" index="251564" value="0"/>
<variable name="ALLOCATE(Z,A066)" index="251565" value="0"/>
<variable name="ALLOCATE(Z,A067)" index="251566" value="0"/>
<variable name="ALLOCATE(Z,A068)" index="251567" value="0"/>
<variable name="ALLOCATE(Z,A069)" index="251568" value="0"/>
<variable name="ALLOCATE(Z,A070)" index="251569" value="0"/>
<variable name="ALLOCATE(Z,A071)" index="251570" value="0"/>
<variable name="ALLOCATE(Z,A072)" index="251571" value="0"/>
<variable name="ALLOCATE(Z,A073)" index="251572" value="0"/>
<variable name="ALLOCATE(Z,A074)" index="251573" value="0"/>
<variable name="ALLOCATE(Z,A075)" index="251574" value="0"/>
<variable name="ALLOCATE(Z,A076)" index="251575" value="0"/>
<variable name="ALLOCATE(Z,A077)" index="251576" value="0"/>
<variable name="ALLOCATE(Z,A078)" index="251577" value="0"/>
<variable name="ALLOCATE(Z,A079)" index="251578" value="0"/>
<variable name="ALLOCATE(Z,A080)" index="251579" value="0"/>
<variable name="ALLOCATE(Z,A081)" index="251580" value="0"/>
<variable name="ALLOCATE(Z,A082)" index="251581" value="0"/>
<variable name="ALLOCATE(Z,A083)" index="251582" value="0"/>
<variable name="ALLOCATE(Z,A084)" index="251583" value="0"/>
<variable name="ALLOCATE(Z,A085)" index="251584" value="0"/>
<variable name="ALLOCATE(Z,A086)" index="251585" value="0"/>
<variable name="ALLOCATE(Z,A087)" index="251586" value="0"/>
<variable name="ALLOCATE(Z,A088)" index="251587" value="0"/>
<variable name="ALLOCATE(Z,A089)" index="251588" value="0"/>
<variable name="ALLOCATE(Z,A090)" index="251589" value="0"/>
<variable name="ALLOCATE(Z,A091)" index="251590" value="0"/>
<variable name="ALLOCATE(Z,A092)" index="251591" value="0"/>
<variable name="ALLOCATE(Z,A093)" index="251592" value="0"/>
<variable name="ALLOCATE(Z,A094)" index="251593" value="0"/>
<variable name="ALLOCATE(Z,A095)" index="251594" value="0"/>
<variable name="ALLOCATE(Z,A096)" index="251595" value="0"/>
<variable name="ALLOCATE(Z,A097)" index="251596" value="0"/>
<variable name="ALLOCATE(Z,A098)" index="251597" value="0"/>
<variable name="ALLOCATE(Z,A099)" index="251598" value="0"/>
<variable name="ALLOCATE(Z,A100)" index="251599" value="0"/>
<variable name="ALLOCATE(Z,B001)" index="251600" value="0"/>
<variable name="ALLOCATE(Z,B002)" index="251601" value="0"/>

<variable name="ALLOCATE(Z,B003)" index="251602" value="0"/>
<variable name="ALLOCATE(Z,B004)" index="251603" value="0"/>
<variable name="ALLOCATE(Z,B005)" index="251604" value="0"/>
<variable name="ALLOCATE(Z,B006)" index="251605" value="0"/>
<variable name="ALLOCATE(Z,B007)" index="251606" value="0"/>
<variable name="ALLOCATE(Z,B008)" index="251607" value="0"/>
<variable name="ALLOCATE(Z,B009)" index="251608" value="1"/>
<variable name="ALLOCATE(Z,B010)" index="251609" value="0"/>
<variable name="ALLOCATE(Z,B011)" index="251610" value="0"/>
<variable name="ALLOCATE(Z,B012)" index="251611" value="0"/>
<variable name="ALLOCATE(Z,B013)" index="251612" value="0"/>
<variable name="ALLOCATE(Z,B014)" index="251613" value="0"/>
<variable name="ALLOCATE(Z,B015)" index="251614" value="0"/>
<variable name="ALLOCATE(Z,B016)" index="251615" value="0"/>
<variable name="ALLOCATE(Z,B017)" index="251616" value="0"/>
<variable name="ALLOCATE(Z,B018)" index="251617" value="1"/>
<variable name="ALLOCATE(Z,B019)" index="251618" value="0"/>
<variable name="ALLOCATE(Z,B020)" index="251619" value="0"/>
<variable name="ALLOCATE(Z,B021)" index="251620" value="0"/>
<variable name="ALLOCATE(Z,B022)" index="251621" value="0"/>
<variable name="ALLOCATE(Z,B023)" index="251622" value="0"/>
<variable name="ALLOCATE(Z,B024)" index="251623" value="0"/>
<variable name="ALLOCATE(Z,B025)" index="251624" value="0"/>
<variable name="ALLOCATE(Z,B026)" index="251625" value="0"/>
<variable name="ALLOCATE(Z,B027)" index="251626" value="0"/>
<variable name="ALLOCATE(Z,B028)" index="251627" value="1"/>
<variable name="ALLOCATE(Z,B029)" index="251628" value="0"/>
<variable name="ALLOCATE(Z,B030)" index="251629" value="0"/>
<variable name="ALLOCATE(Z,B031)" index="251630" value="0"/>
<variable name="ALLOCATE(Z,B032)" index="251631" value="0"/>
<variable name="ALLOCATE(Z,B033)" index="251632" value="0"/>
<variable name="ALLOCATE(Z,B034)" index="251633" value="0"/>
<variable name="ALLOCATE(Z,B035)" index="251634" value="0"/>
<variable name="ALLOCATE(Z,B036)" index="251635" value="0"/>
<variable name="ALLOCATE(Z,B037)" index="251636" value="0"/>
<variable name="ALLOCATE(Z,B038)" index="251637" value="0"/>
<variable name="ALLOCATE(Z,B039)" index="251638" value="0"/>
<variable name="ALLOCATE(Z,B040)" index="251639" value="0"/>
<variable name="ALLOCATE(Z,B041)" index="251640" value="1"/>
<variable name="ALLOCATE(Z,B042)" index="251641" value="0"/>
<variable name="ALLOCATE(Z,B043)" index="251642" value="0"/>
<variable name="ALLOCATE(Z,B044)" index="251643" value="0"/>
<variable name="ALLOCATE(Z,B045)" index="251644" value="0"/>
<variable name="ALLOCATE(Z,B046)" index="251645" value="0"/>
<variable name="ALLOCATE(Z,B047)" index="251646" value="0"/>
<variable name="ALLOCATE(Z,B048)" index="251647" value="0"/>
<variable name="ALLOCATE(Z,B049)" index="251648" value="0"/>
<variable name="ALLOCATE(Z,B050)" index="251649" value="0"/>
<variable name="ALLOCATE(Z,B051)" index="251650" value="0"/>
<variable name="ALLOCATE(Z,B052)" index="251651" value="0"/>
<variable name="ALLOCATE(Z,B053)" index="251652" value="0"/>
<variable name="ALLOCATE(Z,B054)" index="251653" value="0"/>
<variable name="ALLOCATE(Z,B055)" index="251654" value="1"/>
<variable name="ALLOCATE(Z,B056)" index="251655" value="0"/>
<variable name="ALLOCATE(Z,B057)" index="251656" value="0"/>
<variable name="ALLOCATE(Z,B058)" index="251657" value="0"/>
<variable name="ALLOCATE(Z,B059)" index="251658" value="0"/>
<variable name="ALLOCATE(Z,B060)" index="251659" value="0"/>
<variable name="ALLOCATE(Z,B061)" index="251660" value="0"/>
<variable name="ALLOCATE(Z,B062)" index="251661" value="0"/>
<variable name="ALLOCATE(Z,B063)" index="251662" value="0"/>
<variable name="ALLOCATE(Z,B064)" index="251663" value="0"/>
<variable name="ALLOCATE(Z,B065)" index="251664" value="0"/>
<variable name="ALLOCATE(Z,B066)" index="251665" value="0"/>
<variable name="ALLOCATE(Z,B067)" index="251666" value="0"/>
<variable name="ALLOCATE(Z,B068)" index="251667" value="0"/>
<variable name="ALLOCATE(Z,B069)" index="251668" value="0"/>
<variable name="ALLOCATE(Z,B070)" index="251669" value="0"/>
<variable name="ALLOCATE(Z,B071)" index="251670" value="0"/>
<variable name="ALLOCATE(Z,B072)" index="251671" value="0"/>
<variable name="ALLOCATE(Z,B073)" index="251672" value="0"/>
<variable name="ALLOCATE(Z,B074)" index="251673" value="0"/>

<variable name="ALLOCATE(Z,B075)" index="251674" value="0"/>
<variable name="ALLOCATE(Z,B076)" index="251675" value="1"/>
<variable name="ALLOCATE(Z,B077)" index="251676" value="0"/>
<variable name="ALLOCATE(Z,B078)" index="251677" value="0"/>
<variable name="ALLOCATE(Z,B079)" index="251678" value="0"/>
<variable name="ALLOCATE(Z,B080)" index="251679" value="0"/>
<variable name="ALLOCATE(Z,B081)" index="251680" value="0"/>
<variable name="ALLOCATE(Z,B082)" index="251681" value="0"/>
<variable name="ALLOCATE(Z,B083)" index="251682" value="0"/>
<variable name="ALLOCATE(Z,B084)" index="251683" value="0"/>
<variable name="ALLOCATE(Z,B085)" index="251684" value="0"/>
<variable name="ALLOCATE(Z,B086)" index="251685" value="0"/>
<variable name="ALLOCATE(Z,B087)" index="251686" value="0"/>
<variable name="ALLOCATE(Z,B088)" index="251687" value="0"/>
<variable name="ALLOCATE(Z,B089)" index="251688" value="0"/>
<variable name="ALLOCATE(Z,B090)" index="251689" value="0"/>
<variable name="ALLOCATE(Z,B091)" index="251690" value="0"/>
<variable name="ALLOCATE(Z,B092)" index="251691" value="0"/>
<variable name="ALLOCATE(Z,B093)" index="251692" value="0"/>
<variable name="ALLOCATE(Z,B094)" index="251693" value="0"/>
<variable name="ALLOCATE(Z,B095)" index="251694" value="1"/>
<variable name="ALLOCATE(Z,B096)" index="251695" value="0"/>
<variable name="ALLOCATE(Z,B097)" index="251696" value="0"/>
<variable name="ALLOCATE(Z,B098)" index="251697" value="0"/>
<variable name="ALLOCATE(Z,B099)" index="251698" value="0"/>
<variable name="ALLOCATE(Z,B100)" index="251699" value="0"/>
<variable name="ALLOCATE(Z,C001)" index="251700" value="0"/>
<variable name="ALLOCATE(Z,C002)" index="251701" value="0"/>
<variable name="ALLOCATE(Z,C003)" index="251702" value="0"/>
<variable name="ALLOCATE(Z,C004)" index="251703" value="0"/>
<variable name="ALLOCATE(Z,C005)" index="251704" value="0"/>
<variable name="ALLOCATE(Z,C006)" index="251705" value="0"/>
<variable name="ALLOCATE(Z,C007)" index="251706" value="0"/>
<variable name="ALLOCATE(Z,C008)" index="251707" value="0"/>
<variable name="ALLOCATE(Z,C009)" index="251708" value="0"/>
<variable name="ALLOCATE(Z,C010)" index="251709" value="0"/>
<variable name="ALLOCATE(Z,C011)" index="251710" value="0"/>
<variable name="ALLOCATE(Z,C012)" index="251711" value="0"/>
<variable name="ALLOCATE(Z,C013)" index="251712" value="0"/>
<variable name="ALLOCATE(Z,C014)" index="251713" value="0"/>
<variable name="ALLOCATE(Z,C015)" index="251714" value="0"/>
<variable name="ALLOCATE(Z,C016)" index="251715" value="0"/>
<variable name="ALLOCATE(Z,C017)" index="251716" value="0"/>
<variable name="ALLOCATE(Z,C018)" index="251717" value="0"/>
<variable name="ALLOCATE(Z,C019)" index="251718" value="0"/>
<variable name="ALLOCATE(Z,C020)" index="251719" value="0"/>
<variable name="ALLOCATE(Z,C021)" index="251720" value="0"/>
<variable name="ALLOCATE(Z,C022)" index="251721" value="0"/>
<variable name="ALLOCATE(Z,C023)" index="251722" value="0"/>
<variable name="ALLOCATE(Z,C024)" index="251723" value="0"/>
<variable name="ALLOCATE(Z,C025)" index="251724" value="0"/>
<variable name="ALLOCATE(Z,C026)" index="251725" value="0"/>
<variable name="ALLOCATE(Z,C027)" index="251726" value="0"/>
<variable name="ALLOCATE(Z,C028)" index="251727" value="0"/>
<variable name="ALLOCATE(Z,C029)" index="251728" value="0"/>
<variable name="ALLOCATE(Z,C030)" index="251729" value="0"/>
<variable name="ALLOCATE(Z,C031)" index="251730" value="0"/>
<variable name="ALLOCATE(Z,C032)" index="251731" value="0"/>
<variable name="ALLOCATE(Z,C033)" index="251732" value="0"/>
<variable name="ALLOCATE(Z,C034)" index="251733" value="0"/>
<variable name="ALLOCATE(Z,C035)" index="251734" value="0"/>
<variable name="ALLOCATE(Z,C036)" index="251735" value="0"/>
<variable name="ALLOCATE(Z,C037)" index="251736" value="0"/>
<variable name="ALLOCATE(Z,C038)" index="251737" value="0"/>
<variable name="ALLOCATE(Z,C039)" index="251738" value="0"/>
<variable name="ALLOCATE(Z,C040)" index="251739" value="0"/>
<variable name="ALLOCATE(Z,C041)" index="251740" value="0"/>
<variable name="ALLOCATE(Z,C042)" index="251741" value="0"/>
<variable name="ALLOCATE(Z,C043)" index="251742" value="0"/>
<variable name="ALLOCATE(Z,C044)" index="251743" value="0"/>
<variable name="ALLOCATE(Z,C045)" index="251744" value="0"/>
<variable name="ALLOCATE(Z,C046)" index="251745" value="0"/>

```
<variable name="ALLOCATE(Z,C047)" index="251746" value="0"/>
<variable name="ALLOCATE(Z,C048)" index="251747" value="0"/>
<variable name="ALLOCATE(Z,C049)" index="251748" value="0"/>
<variable name="ALLOCATE(Z,C050)" index="251749" value="0"/>
<variable name="ALLOCATE(Z,C051)" index="251750" value="0"/>
<variable name="ALLOCATE(Z,C052)" index="251751" value="0"/>
<variable name="ALLOCATE(Z,C053)" index="251752" value="0"/>
<variable name="ALLOCATE(Z,C054)" index="251753" value="0"/>
<variable name="ALLOCATE(Z,C055)" index="251754" value="0"/>
<variable name="ALLOCATE(Z,C056)" index="251755" value="0"/>
<variable name="ALLOCATE(Z,C057)" index="251756" value="0"/>
<variable name="ALLOCATE(Z,C058)" index="251757" value="0"/>
<variable name="ALLOCATE(Z,C059)" index="251758" value="0"/>
<variable name="ALLOCATE(Z,C060)" index="251759" value="0"/>
<variable name="ALLOCATE(Z,C061)" index="251760" value="0"/>
<variable name="ALLOCATE(Z,C062)" index="251761" value="0"/>
<variable name="ALLOCATE(Z,C063)" index="251762" value="0"/>
<variable name="ALLOCATE(Z,C064)" index="251763" value="0"/>
<variable name="ALLOCATE(Z,C065)" index="251764" value="0"/>
<variable name="ALLOCATE(Z,C066)" index="251765" value="0"/>
<variable name="ALLOCATE(Z,C067)" index="251766" value="0"/>
<variable name="ALLOCATE(Z,C068)" index="251767" value="0"/>
<variable name="ALLOCATE(Z,C069)" index="251768" value="0"/>
<variable name="ALLOCATE(Z,C070)" index="251769" value="0"/>
<variable name="ALLOCATE(Z,C071)" index="251770" value="0"/>
<variable name="ALLOCATE(Z,C072)" index="251771" value="0"/>
<variable name="ALLOCATE(Z,C073)" index="251772" value="0"/>
<variable name="ALLOCATE(Z,C074)" index="251773" value="1"/>
<variable name="ALLOCATE(Z,C075)" index="251774" value="0"/>
<variable name="ALLOCATE(Z,C076)" index="251775" value="0"/>
<variable name="ALLOCATE(Z,C077)" index="251776" value="1"/>
<variable name="ALLOCATE(Z,C078)" index="251777" value="0"/>
<variable name="ALLOCATE(Z,C079)" index="251778" value="1"/>
<variable name="ALLOCATE(Z,C080)" index="251779" value="0"/>
<variable name="ALLOCATE(Z,C081)" index="251780" value="0"/>
<variable name="ALLOCATE(Z,C082)" index="251781" value="0"/>
<variable name="ALLOCATE(Z,C083)" index="251782" value="0"/>
<variable name="ALLOCATE(Z,C084)" index="251783" value="0"/>
<variable name="ALLOCATE(Z,C085)" index="251784" value="0"/>
<variable name="ALLOCATE(Z,C086)" index="251785" value="0"/>
<variable name="ALLOCATE(Z,C087)" index="251786" value="0"/>
<variable name="ALLOCATE(Z,C088)" index="251787" value="1"/>
<variable name="ALLOCATE(Z,C089)" index="251788" value="1"/>
<variable name="ALLOCATE(Z,C090)" index="251789" value="0"/>
<variable name="ALLOCATE(Z,C091)" index="251790" value="0"/>
<variable name="ALLOCATE(Z,C092)" index="251791" value="0"/>
<variable name="ALLOCATE(Z,C093)" index="251792" value="0"/>
<variable name="ALLOCATE(Z,C094)" index="251793" value="0"/>
<variable name="ALLOCATE(Z,C095)" index="251794" value="1"/>
<variable name="ALLOCATE(Z,C096)" index="251795" value="0"/>
<variable name="ALLOCATE(Z,C097)" index="251796" value="0"/>
<variable name="ALLOCATE(Z,C098)" index="251797" value="0"/>
<variable name="ALLOCATE(Z,C099)" index="251798" value="0"/>
<variable name="ALLOCATE(Z,C100)" index="251799" value="0"/>
<variable name="ALLOCATE(Z,D001)" index="251800" value="0"/>
<variable name="ALLOCATE(Z,D002)" index="251801" value="0"/>
<variable name="ALLOCATE(Z,D003)" index="251802" value="0"/>
<variable name="ALLOCATE(Z,D004)" index="251803" value="0"/>
<variable name="ALLOCATE(Z,D005)" index="251804" value="0"/>
<variable name="ALLOCATE(Z,D006)" index="251805" value="0"/>
<variable name="ALLOCATE(Z,D007)" index="251806" value="0"/>
<variable name="ALLOCATE(Z,D008)" index="251807" value="0"/>
<variable name="ALLOCATE(Z,D009)" index="251808" value="0"/>
<variable name="ALLOCATE(Z,D010)" index="251809" value="0"/>
<variable name="ALLOCATE(Z,D011)" index="251810" value="0"/>
<variable name="ALLOCATE(Z,D012)" index="251811" value="0"/>
<variable name="ALLOCATE(Z,D013)" index="251812" value="0"/>
<variable name="ALLOCATE(Z,D014)" index="251813" value="0"/>
<variable name="ALLOCATE(Z,D015)" index="251814" value="0"/>
<variable name="ALLOCATE(Z,D016)" index="251815" value="0"/>
<variable name="ALLOCATE(Z,D017)" index="251816" value="0"/>
<variable name="ALLOCATE(Z,D018)" index="251817" value="0"/>
```

<variable name="ALLOCATE(Z,D019)" index="251818" value="0"/>
<variable name="ALLOCATE(Z,D020)" index="251819" value="0"/>
<variable name="ALLOCATE(Z,D021)" index="251820" value="0"/>
<variable name="ALLOCATE(Z,D022)" index="251821" value="0"/>
<variable name="ALLOCATE(Z,D023)" index="251822" value="0"/>
<variable name="ALLOCATE(Z,D024)" index="251823" value="0"/>
<variable name="ALLOCATE(Z,D025)" index="251824" value="0"/>
<variable name="ALLOCATE(Z,D026)" index="251825" value="1"/>
<variable name="ALLOCATE(Z,D027)" index="251826" value="0"/>
<variable name="ALLOCATE(Z,D028)" index="251827" value="1"/>
<variable name="ALLOCATE(Z,D029)" index="251828" value="0"/>
<variable name="ALLOCATE(Z,D030)" index="251829" value="0"/>
<variable name="ALLOCATE(Z,D031)" index="251830" value="0"/>
<variable name="ALLOCATE(Z,D032)" index="251831" value="0"/>
<variable name="ALLOCATE(Z,D033)" index="251832" value="0"/>
<variable name="ALLOCATE(Z,D034)" index="251833" value="0"/>
<variable name="ALLOCATE(Z,D035)" index="251834" value="0"/>
<variable name="ALLOCATE(Z,D036)" index="251835" value="0"/>
<variable name="ALLOCATE(Z,D037)" index="251836" value="0"/>
<variable name="ALLOCATE(Z,D038)" index="251837" value="0"/>
<variable name="ALLOCATE(Z,D039)" index="251838" value="0"/>
<variable name="ALLOCATE(Z,D040)" index="251839" value="0"/>
<variable name="ALLOCATE(Z,D041)" index="251840" value="0"/>
<variable name="ALLOCATE(Z,D042)" index="251841" value="0"/>
<variable name="ALLOCATE(Z,D043)" index="251842" value="0"/>
<variable name="ALLOCATE(Z,D044)" index="251843" value="0"/>
<variable name="ALLOCATE(Z,D045)" index="251844" value="0"/>
<variable name="ALLOCATE(Z,D046)" index="251845" value="0"/>
<variable name="ALLOCATE(Z,D047)" index="251846" value="0"/>
<variable name="ALLOCATE(Z,D048)" index="251847" value="0"/>
<variable name="ALLOCATE(Z,D049)" index="251848" value="0"/>
<variable name="ALLOCATE(Z,D050)" index="251849" value="0"/>
<variable name="ALLOCATE(Z,D051)" index="251850" value="0"/>
<variable name="ALLOCATE(Z,D052)" index="251851" value="0"/>
<variable name="ALLOCATE(Z,D053)" index="251852" value="0"/>
<variable name="ALLOCATE(Z,D054)" index="251853" value="0"/>
<variable name="ALLOCATE(Z,D055)" index="251854" value="0"/>
<variable name="ALLOCATE(Z,D056)" index="251855" value="0"/>
<variable name="ALLOCATE(Z,D057)" index="251856" value="0"/>
<variable name="ALLOCATE(Z,D058)" index="251857" value="0"/>
<variable name="ALLOCATE(Z,D059)" index="251858" value="0"/>
<variable name="ALLOCATE(Z,D060)" index="251859" value="1"/>
<variable name="ALLOCATE(Z,D061)" index="251860" value="0"/>
<variable name="ALLOCATE(Z,D062)" index="251861" value="0"/>
<variable name="ALLOCATE(Z,D063)" index="251862" value="0"/>
<variable name="ALLOCATE(Z,D064)" index="251863" value="0"/>
<variable name="ALLOCATE(Z,D065)" index="251864" value="1"/>
<variable name="ALLOCATE(Z,D066)" index="251865" value="0"/>
<variable name="ALLOCATE(Z,D067)" index="251866" value="0"/>
<variable name="ALLOCATE(Z,D068)" index="251867" value="0"/>
<variable name="ALLOCATE(Z,D069)" index="251868" value="0"/>
<variable name="ALLOCATE(Z,D070)" index="251869" value="0"/>
<variable name="ALLOCATE(Z,D071)" index="251870" value="0"/>
<variable name="ALLOCATE(Z,D072)" index="251871" value="0"/>
<variable name="ALLOCATE(Z,D073)" index="251872" value="0"/>
<variable name="ALLOCATE(Z,D074)" index="251873" value="0"/>
<variable name="ALLOCATE(Z,D075)" index="251874" value="0"/>
<variable name="ALLOCATE(Z,D076)" index="251875" value="0"/>
<variable name="ALLOCATE(Z,D077)" index="251876" value="0"/>
<variable name="ALLOCATE(Z,D078)" index="251877" value="0"/>
<variable name="ALLOCATE(Z,D079)" index="251878" value="0"/>
<variable name="ALLOCATE(Z,D080)" index="251879" value="0"/>
<variable name="ALLOCATE(Z,D081)" index="251880" value="0"/>
<variable name="ALLOCATE(Z,D082)" index="251881" value="0"/>
<variable name="ALLOCATE(Z,D083)" index="251882" value="0"/>
<variable name="ALLOCATE(Z,D084)" index="251883" value="0"/>
<variable name="ALLOCATE(Z,D085)" index="251884" value="0"/>
<variable name="ALLOCATE(Z,D086)" index="251885" value="0"/>
<variable name="ALLOCATE(Z,D087)" index="251886" value="1"/>
<variable name="ALLOCATE(Z,D088)" index="251887" value="0"/>
<variable name="ALLOCATE(Z,D089)" index="251888" value="0"/>
<variable name="ALLOCATE(Z,D090)" index="251889" value="0"/>

<variable name="ALLOCATE(Z,D091)" index="251890" value="0"/>
<variable name="ALLOCATE(Z,D092)" index="251891" value="0"/>
<variable name="ALLOCATE(Z,D093)" index="251892" value="0"/>
<variable name="ALLOCATE(Z,D094)" index="251893" value="1"/>
<variable name="ALLOCATE(Z,D095)" index="251894" value="0"/>
<variable name="ALLOCATE(Z,D096)" index="251895" value="0"/>
<variable name="ALLOCATE(Z,D097)" index="251896" value="0"/>
<variable name="ALLOCATE(Z,D098)" index="251897" value="0"/>
<variable name="ALLOCATE(Z,D099)" index="251898" value="0"/>
<variable name="ALLOCATE(Z,D100)" index="251899" value="0"/>
<variable name="ALLOCATE(Z,E001)" index="251900" value="0"/>
<variable name="ALLOCATE(Z,E002)" index="251901" value="0"/>
<variable name="ALLOCATE(Z,E003)" index="251902" value="1"/>
<variable name="ALLOCATE(Z,E004)" index="251903" value="0"/>
<variable name="ALLOCATE(Z,E005)" index="251904" value="0"/>
<variable name="ALLOCATE(Z,E006)" index="251905" value="0"/>
<variable name="ALLOCATE(Z,E007)" index="251906" value="0"/>
<variable name="ALLOCATE(Z,E008)" index="251907" value="0"/>
<variable name="ALLOCATE(Z,E009)" index="251908" value="0"/>
<variable name="ALLOCATE(Z,E010)" index="251909" value="0"/>
<variable name="ALLOCATE(Z,E011)" index="251910" value="0"/>
<variable name="ALLOCATE(Z,E012)" index="251911" value="0"/>
<variable name="ALLOCATE(Z,E013)" index="251912" value="0"/>
<variable name="ALLOCATE(Z,E014)" index="251913" value="0"/>
<variable name="ALLOCATE(Z,E015)" index="251914" value="0"/>
<variable name="ALLOCATE(Z,E016)" index="251915" value="0"/>
<variable name="ALLOCATE(Z,E017)" index="251916" value="0"/>
<variable name="ALLOCATE(Z,E018)" index="251917" value="1"/>
<variable name="ALLOCATE(Z,E019)" index="251918" value="0"/>
<variable name="ALLOCATE(Z,E020)" index="251919" value="0"/>
<variable name="ALLOCATE(Z,E021)" index="251920" value="0"/>
<variable name="ALLOCATE(Z,E022)" index="251921" value="0"/>
<variable name="ALLOCATE(Z,E023)" index="251922" value="0"/>
<variable name="ALLOCATE(Z,E024)" index="251923" value="1"/>
<variable name="ALLOCATE(Z,E025)" index="251924" value="1"/>
<variable name="ALLOCATE(Z,E026)" index="251925" value="0"/>
<variable name="ALLOCATE(Z,E027)" index="251926" value="0"/>
<variable name="ALLOCATE(Z,E028)" index="251927" value="0"/>
<variable name="ALLOCATE(Z,E029)" index="251928" value="0"/>
<variable name="ALLOCATE(Z,E030)" index="251929" value="0"/>
<variable name="ALLOCATE(Z,E031)" index="251930" value="0"/>
<variable name="ALLOCATE(Z,E032)" index="251931" value="0"/>
<variable name="ALLOCATE(Z,E033)" index="251932" value="0"/>
<variable name="ALLOCATE(Z,E034)" index="251933" value="0"/>
<variable name="ALLOCATE(Z,E035)" index="251934" value="0"/>
<variable name="ALLOCATE(Z,E036)" index="251935" value="0"/>
<variable name="ALLOCATE(Z,E037)" index="251936" value="0"/>
<variable name="ALLOCATE(Z,E038)" index="251937" value="0"/>
<variable name="ALLOCATE(Z,E039)" index="251938" value="0"/>
<variable name="ALLOCATE(Z,E040)" index="251939" value="0"/>
<variable name="ALLOCATE(Z,E041)" index="251940" value="0"/>
<variable name="ALLOCATE(Z,E042)" index="251941" value="0"/>
<variable name="ALLOCATE(Z,E043)" index="251942" value="0"/>
<variable name="ALLOCATE(Z,E044)" index="251943" value="0"/>
<variable name="ALLOCATE(Z,E045)" index="251944" value="0"/>
<variable name="ALLOCATE(Z,E046)" index="251945" value="0"/>
<variable name="ALLOCATE(Z,E047)" index="251946" value="0"/>
<variable name="ALLOCATE(Z,E048)" index="251947" value="0"/>
<variable name="ALLOCATE(Z,E049)" index="251948" value="0"/>
<variable name="ALLOCATE(Z,E050)" index="251949" value="0"/>
<variable name="ALLOCATE(Z,E051)" index="251950" value="0"/>
<variable name="ALLOCATE(Z,E052)" index="251951" value="0"/>
<variable name="ALLOCATE(Z,E053)" index="251952" value="0"/>
<variable name="ALLOCATE(Z,E054)" index="251953" value="0"/>
<variable name="ALLOCATE(Z,E055)" index="251954" value="0"/>
<variable name="ALLOCATE(Z,E056)" index="251955" value="0"/>
<variable name="ALLOCATE(Z,E057)" index="251956" value="0"/>
<variable name="ALLOCATE(Z,E058)" index="251957" value="0"/>
<variable name="ALLOCATE(Z,E059)" index="251958" value="0"/>
<variable name="ALLOCATE(Z,E060)" index="251959" value="0"/>
<variable name="ALLOCATE(Z,E061)" index="251960" value="0"/>
<variable name="ALLOCATE(Z,E062)" index="251961" value="0"/>

```
<variable name="ALLOCATE(Z,E063)" index="251962" value="0"/>
<variable name="ALLOCATE(Z,E064)" index="251963" value="0"/>
<variable name="ALLOCATE(Z,E065)" index="251964" value="0"/>
<variable name="ALLOCATE(Z,E066)" index="251965" value="1"/>
<variable name="ALLOCATE(Z,E067)" index="251966" value="0"/>
<variable name="ALLOCATE(Z,E068)" index="251967" value="0"/>
<variable name="ALLOCATE(Z,E069)" index="251968" value="0"/>
<variable name="ALLOCATE(Z,E070)" index="251969" value="0"/>
<variable name="ALLOCATE(Z,E071)" index="251970" value="0"/>
<variable name="ALLOCATE(Z,E072)" index="251971" value="0"/>
<variable name="ALLOCATE(Z,E073)" index="251972" value="1"/>
<variable name="ALLOCATE(Z,E074)" index="251973" value="0"/>
<variable name="ALLOCATE(Z,E075)" index="251974" value="0"/>
<variable name="ALLOCATE(Z,E076)" index="251975" value="0"/>
<variable name="ALLOCATE(Z,E077)" index="251976" value="0"/>
<variable name="ALLOCATE(Z,E078)" index="251977" value="0"/>
<variable name="ALLOCATE(Z,E079)" index="251978" value="0"/>
<variable name="ALLOCATE(Z,E080)" index="251979" value="0"/>
<variable name="ALLOCATE(Z,E081)" index="251980" value="0"/>
<variable name="ALLOCATE(Z,E082)" index="251981" value="0"/>
<variable name="ALLOCATE(Z,E083)" index="251982" value="0"/>
<variable name="ALLOCATE(Z,E084)" index="251983" value="0"/>
<variable name="ALLOCATE(Z,E085)" index="251984" value="1"/>
<variable name="ALLOCATE(Z,E086)" index="251985" value="0"/>
<variable name="ALLOCATE(Z,E087)" index="251986" value="0"/>
<variable name="ALLOCATE(Z,E088)" index="251987" value="0"/>
<variable name="ALLOCATE(Z,E089)" index="251988" value="0"/>
<variable name="ALLOCATE(Z,E090)" index="251989" value="0"/>
<variable name="ALLOCATE(Z,E091)" index="251990" value="0"/>
<variable name="ALLOCATE(Z,E092)" index="251991" value="0"/>
<variable name="ALLOCATE(Z,E093)" index="251992" value="0"/>
<variable name="ALLOCATE(Z,E094)" index="251993" value="0"/>
<variable name="ALLOCATE(Z,E095)" index="251994" value="1"/>
<variable name="ALLOCATE(Z,E096)" index="251995" value="0"/>
<variable name="ALLOCATE(Z,E097)" index="251996" value="0"/>
<variable name="ALLOCATE(Z,E098)" index="251997" value="0"/>
<variable name="ALLOCATE(Z,E099)" index="251998" value="0"/>
<variable name="ALLOCATE(Z,E100)" index="251999" value="0"/>
```